Nicola Blefari-Melazzi
Giuseppe Bianchi
Luca Salgarelli *Editors*

Trustworthy Internet

Springer

Trustworthy Internet

Nicola Blefari-Melazzi
Giuseppe Bianchi · Luca Salgarelli
Editors

Trustworthy Internet

Springer

Editors
Nicola Blefari-Melazzi
Department of Electronic Engineering
University of Rome "Tor Vergata"
Via del Politecnico 1
00133 Rome
Italy
e-mail: blefari@uniroma2.it

Luca Salgarelli
Department of Information Engineering
University of Brescia
Via Branze 38
25123 Brescia
Italy
e-mail: luca.salgarelli@ing.unibs.it

Giuseppe Bianchi
Department of Electronic Engineering
University of Rome "Tor Vergata"
Via del Politecnico 1
00133 Rome
Italy
e-mail: giuseppe.bianchi@uniroma2.it

Selected contributions from the 2010 International Tyrrhenian Workshop on Digital Communication

ISBN 978-88-470-1817-4 e-ISBN 978-88-470-1818-1

DOI 10.1007/978-88-470-1818-1

Springer Milan Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010937688

Cover design: eStudio Calamar S.L.

Printed on acid-free paper, June 2011

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

This book aims at providing a snapshot of the various facets (and underlying challenges) that contribute to make the Internet trustworthy. It originated from the 21st International Tyrrhenian Workshop on Digital Communications, an event traditionally organized by CNIT, the Italian inter-university consortium for telecommunication research. The workshop is devoted each year to a specific theme in the area of digital communications and its applications, and the theme selected for this year's edition was "Trustworthy Internet" (tyrr2010.cnit.it), which the European Union's research program defines as a network characterized by the following properties: secure, reliable and resilient to attacks and operational failures; guaranteeing quality of service; protecting user data; ensuring privacy and providing usable and trusted tools to support the users in their security management.

The workshop program comprised both peer-reviewed papers and invited contributions from leading experts in the field. It provided a lively and exciting discussion on the technical challenges and issues involved in the trustworthy rethinking of the Future Internet. This book aims at offering the findings and discussions carried out during the workshop days to a broader audience. For this purpose, the book contains a selection of the works presented at the 21st International Tyrrhenian Workshop on Digital Communications. Each contribution has been extended with background material on the specific topic in question. Moreover, the sections account for the supplementary insights gathered from the workshop discussions, and, when appropriate, also include additional technical results.

We thank the persons who have been instrumental in making our workshop a success, and who have permitted this book to exist: the contributors for their high quality works and their supplementary effort spent in making their findings accessible to the broader audience expected for a book, the reviewers of the papers for their thorough work; the speakers, the session chairs and the audience for attending and making the sessions a lively and fruitful environment; the organizers of the invited sessions, who also contributed sustantially to the Introduction and organisation of this book: Dr. Thorsten Strufe, Dr. Sonja Buchegger, Prof. Fabio

Massacci, Dr. Federica Paci, Dr. Saverio Niccolini, Dr. Sandra Tartarelli, Dr. Leonardo Chiariglione; a special thank to Leonardo who gave also the keynote speech: "*Catering to the sustainable evolution of the digital media foundations*".

We are also grateful to our main sponsor Confcommercio, and its President, Dr. Carlo Sangalli, for the generous support in such a difficult time. Indeed, we are proud that such an highly recognized Italian institution has tangibly demonstrated a strong interest in sustaining research efforts targeting a more secure, efficient, and trusted Internet environment.

We thank the research projects belonging to the 7th Framework Programme of the European Union in which CNIT is involved, and working in areas relevant to the workshop theme, for providing technical support: CONVERGENCE, DEMONS, FLAVIA, MOMENT, PERIMETER, PRISM, with a special thank to the CONVERGENCE project that supported the workshop also with additional funding.

And we finally thank the CNIT president, Prof. Giancarlo Prati, the CNIT Director, Prof. Silvano Pupolin, and the CNIT executive board, for having given us the opportunity to organize this workshop on the very timely and important topic of trustworthy Internet.

N. Blefari Melazzi
G. Bianchi
L. Salgarelli

Introduction

The term "trustworthy" is explicitly defined by the European Community's FP7 research program as: secure, reliable and resilient to attacks and operational failures; guaranteeing quality of service; protecting user data; ensuring privacy and providing usable and trusted tools to support the user in his security management. As such, the Trustworthy Internet not only has to include mechanisms, architectures and networking infrastructures that intrinsically provide basic security guarantees, but it also has to ensure users, service providers and application providers alike that their requirements in terms of Quality of Experience, manageability and efficiency are fully met. Providing such combined guarantees in a rapidly evolving, complex infrastructure such as the Internet requires solving challenging issues that encompass many fields of theoretical and applied information engineering, spanning all levels of the protocol stack, ranging from finding new intrinsically secure transmission systems, to radically novel routing models, to new architectures for data dissemination and for interconnecting an unprecedented number of devices and appliances.

This book aims at representing a view of the state of the "Trustworthy Internet" as we enter the second decade of our century. The material included in this book originates from a workshop, organized in September 2010, and specifically dedicated to the several aspects which contribute to make the today's and tomorrow's Internet trustworthy. The workshop comprised either invited contributions from renowned researchers with complementary expertise, as well as independent, peer-reviewed contributions stimulated through an open call for papers. The book includes a selected subset of the workshop papers. Each contribution has been edited and extended after the workshop, taking into account the discussions carried out during the event, incorporating when appropriate additional technical material, and, perhaps most importantly, complementing the specific technical aspects presented with background material devised to more comprehensively introduce the reader to the specific topic of trustworthiness tackled.

For the reader's convenience, the selected scientific contributions are further grouped in homogeneous chapters that present scholarly visions of many of the

aspects defined by the term *trustworthy*. As such, it is our hope that this material can serve scholars, practitioners and students alike as a guide not only to help understanding the state of current research in this topics, but can also represent a guided look at its medium term future.

The material in *Part I* focuses on the research that aims at imagining how *the future Internet will support trustworthiness*. Although predicting the future is always risky business, it is clear that the Internet is changing quickly, starting from the communication paradigms that are shifting rapidly from the client-server model to more advanced ones, such as publish-subscribe, overlay and community-oriented networking. These new communication models not only pose interesting and novel challenges to operational aspects of networking, such as efficiently supporting scalable QoS requirements and dealing with fast routing in heterogeneous networks, but also stimulate the research of new mechanisms to support the security and trust of such new infrastructures. Contributions in this chapter revolve around the last aspect.

Chapter 1 discusses the security and privacy properties of a new prospective Internet architecture based on the publish-subscribe paradigm. Indeed, the information oriented nature of publish-subscribe, the decoupling it offers between information providers and information consumers, its location-identity split properties, and the clean-slate ambition of replacing the actual Internet protocol stack rather than complementing it, calls for specifically tailored security and privacy features, and opens up a number of key challenges which are duly presented and discussed in this section. Chapter 2 looks at security aspects in programmable, high-performance routers, which are basic building blocks for network resource virtualization and dynamic service configuration in future Internet scenarios. Router programmability requires a careful scrutiny of the security aspects which arise when either customers as well as service providers access the routing nodes. This chapter specifically discusses promising solutions to the support of secure services through the separation of control and forwarding elements.

Identity management, i.e. governing the fine relationship between privacy and trust, in overlay networks is explored by the last two chapters of this Part. Chapter 3 looks at the problem of how to provide secure pseudonymous access from the perspective of an overlay data dissemination network. Chapter 4 attacks the problem of building a wide-area, application-layer identity management for a Semantic Web architecture aimed at supporting seamless data interoperability to facilitate sharing and exploitation of distributed and heterogeneous user profile information.

Part II overviews specific trustworthiness issues, tools, methodologies and their applications to different aspects of the current Internet architecture. Chapter 5 surveys context privacy protection approaches and solutions for the Internet of Things, which represents a challenging issue given the scalability requirements of the sensory world. This chapter specifically deals with the various facets of privacy issues, here categorized as Application privacy, concerning the protection of the network application being run, Correspondent privacy, concerning identity

protection of the other peer in a communication, and Location privacy, concerning the protection of the user location information. Chapter 6 first discusses emerging and novel ways for providing security directly at the physical layer, and then provides a thorough overview of a new modulation technique, able to encrypt the radio signal without any a priori common secret between the two nodes. Here, the information is modulated, at physical layer, by the thermal noise experienced by the link between two terminals, and a loop scheme is designed for the recovery of mutual information. Chapter 7 introduces the reader to the important area of secure multiparty computation. While showing a special case of privacy-preserving computation, the chapter permits the reader to get acquainted with the application of homomorphic encryption to the computation of vector multiplications, scalar products, renormalizations, etc. The tools and methodology here described have a wide applicability in many fields, and show that non elementary operations can be performed without impairing the privacy of the data. Chapter 8 presents a fully distributed security framework for Peer-to-peer Voice over IP, which does not rely on a centralized PKI, but leverages and adapts brilliant approaches, such as ZRTP from Zimmermann, which are deemed to significantly influence future Internet community/distributed services.

Finally, the last two contributions tackle, for different contexts (4G femto cells and ZigBee sensor networks), the issue of spontaneous, autonomic configuration. Specifically, Chapter 9 takes a look at novel ways of dynamically sharing a scarce resource, i.e., the spectrum, so as to provide QoS guarantees in wireless environments with high density of low coverage wireless femto-cells, such as the ones typical of home or community networks. The solutions presented do not require coordination, and may be applied also to the futuristic (and here shown to bring performance advantages) scenario of *different* operators sharing a same subset of frequency bands. Finally, Chapter 10 uses network monitoring information for automatically re-configuring autonomic, self-organizing wireless personal area networks in an energy-aware and robust manner.

Part III deals with the topics related to Online Social Networks, which are increasingly popular with Internet users. They attract ever-growing groups of participants, who in addition spend increasing amounts of time on these sites. Facebook, the currently largest Online Social Network, alone claims an active user base of over 550 million individuals. Both with respect to the number of available pages and served bandwidth it constantly ranks among the top three web sites worldwide (see for example http://www.hitwise.com). Considering the time the users spend on a web brand's pages, it has surpassed the former top competitor, Google and its affiliated services, with users spending almost three times as long on Facebook's, compared to Google and its affiliates' pages (over seven hours per month on Facebook compared to just under 2.5 hours on Google's sites in January 2010, see "Top web sites and Brands" at The Nielsen Company). In addition to user time spent on online social networks, there is an increase in personal information revealed online. Users of Online Social Networks upon registration create a profile describing themselves and they henceforth can (and are expected to) connect their profile with links to the profiles of their friends, family, and

acquaintances. Online Social Networks thus represent a subset of the relations their users entertain in the real world. They additionally offer rich functionality to publish and share various content, to communicate by different means, and to collaborate and play games together. Online Social Networks by nature contain a wealth of personally identifiable information, since each published item and each message inherently is linked to the personal profile identifying the publisher or sender.

Collaborative games in Online Social Networks frequently leverage a player's social environment: players do not play alone or in predetermined groups, but the users added in the contact list of the player automatically act as partners or supporters in the game. This causes a change in the perception of friendship and trust, since contacts are made for the sole reason of two strangers wanting to support each other in one of the games. Chapter 11 deals with this issue and the consequences for trust and privacy from the security perspective. The highly accurate and authentic identification of users comes with both beneficial and adverse consequences. Recognition and reputation now can be attributed to individuals and have a direct effect on real life, as opposed to the reputation of anonymous clients, devices, or personae before. Profiling and targeting users is performed directly on individuals as well. This conflict is the topic of Chapter 12, where the field of participatory sensing is described, and new methods are proposed to leverage the positive social effects, while using abstraction to mitigate the consequences for the privacy of the participants. The voluntarily self-maintained and easily exploitable database of personally identifiable information that Online Social Networks represent is of high value to third parties. The possibility to create highly detailed and perfectly identified behavioral profiles of individuals is not only very attractive to the advertisement industry, but also to miscreants with various adverse motivations. This situation is the focus of Chapter 13, where several attacks are described that are possible and additionally exceptionally successful due to the social and seemingly trustable environment that Online Social Networks provide. Finally, Chapter 14 motivates the decentralization of Online Social Networks for the purpose of load balancing and, more importantly, for the protection of the users' privacy. This chapter subsequently presents a survey of the current proposals for decentralized Online Social Networks and classifies them by characteristic properties.

New technologies and tools including Web services, blogs, social network sites, mash ups, wikis have dramatically changed the way users communicate, collaborate and share personal information on the Internet.

Part IV groups contributions that present the various facets of trustworthyness in Web 2.0 platforms. The main focus here is on finding mechanisms able to guarantee that the composition of services satisfies certain security properties, while maintaining the system flexible and scalable. Web service technology allows users to pull together content from different sources and services to build a new service. Such technology has also facilitated the collaboration intra- and inter-organizations by making accessible organizational business processes through the Web. Blogs, wikis and social networking sites allow users to share new content

and to collaborate and communicate with others. The power of Web 2.0 brings up a number of serious security issues ranging from identity management and reputation, privacy protection and anonymous usage, access control to content and services to integrity protection of composite services. Securing Web 2.0 applications is challenging because such applications are targeted towards making people, information, and resources available to all who need it, and to adapt swiftly to changes and evolutions in software platform and usage models. In contrast, information security seeks to restrict access only to those with proper authorization.

Chapter 15 presents a broad survey of the many challenges faced by the mechanisms used to protect social networks, where content is dynamically updated and accessed by millions of users on a daily basis. In this scenario we must be able to enforce access control policies on social network users' profiles that meet the privacy preferences of the users. This chapter discusses the various alternatives based on risk, trust and other metrics and the related trade-off. The other three chapters discuss more in details the impact of the evolution of data types, user roles and usage models on the security solutions for Web 2.0 applications. Chapter 16 deals with the problem of managing the temporal evolution of authorizations in which users' attributes can change over time and in particular can change after access has been granted. The presented model is an extension of the (now) classical usage control model by Park and Sandhu: it guarantees that access control policies are enforced not only when a resource is accessed but also during its usage. Chapter 17 investigates the challenges of selecting services that meet specific privacy and security properties to build complex composite applications such as business processes. In particular, the problem of dynamic certification of composite services is analyzed, starting from the composition of the certificates of the composite services. Chapter 18 analyzes the same problem but from the perspective of usability. While it is true that users can change attributes as time goes by, the goals of an organization or the objectives of the individuals in a social or collaborative environments are more stable. We would like to be able to achieve them even in presence of changed security circumstances. This chapter discusses the problem of dynamic resiliency of business processes, that is how to guarantee that business processes can still be executed when users authorizations change and that overall business goals can be achieved by suitably re-distributing the authorizations to the appropriate users.

Part V focuses on various facets of network monitoring with a particular attention to trustworthiness applications. Indeed, network monitoring is an area which is expected to face a number of radical changes in the near future. Security threats, which once represented mere "hacking" or exploitation of hosts for little more than curiosity or vanity, have given way to sophisticated criminal operations that exploit vulnerabilities in network devices and end systems to take over large numbers of nodes, arranging them into botnets, for spamming, phishing, extortion via distributed denial of service attacks, and personal information theft, threatening end-user privacy and the importance of "information as an asset". To make matters more challenging, there is an ongoing trend towards bringing Internet

technologies into every end device. A very high number of nodes (e.g., every phone on a network, every device in a household, etc.) are now becoming IP-enabled, more intelligent and more complex, providing new hosts for botnets and bridging legacy and IP-based systems. This poses serious challenges to the operators, as the problems multiply while the requirement on trustworthiness remains unchanged. In addition, monitoring systems must be scalable. Internet traffic growth is reaching volumes previously unimagined. Annual global IP traffic volume nearly doubles every two years, and will exceed half a zettabyte (5×1020 bytes) by 2012. This growth poses severe challenges to the Internet scalability, and calls for a decentralized and scalable monitoring infrastructure. Furthermore, monitoring infrastructures must take privacy into account. Indeed, traffic monitoring activities, especially at higher layers of the network stack, pose a serious risk to the privacy of the individual, since they may result in tracking the personal activities of the end users without their knowledge. Monitoring activities undertaken without transparency or accountability with respect to data processing, i.e. without privacy-awareness, lead to a loss of trust in the network as a whole. As a result, care must be taken that privacy concerns are addressed, and that privacy rights and data protection laws are not violated. And, finally, network monitoring applications are moving up the layers. Traditional monitoring applications are no more focused on the analysis of IP-level traffic but instead tend to include more and more application-specific information and semantics in order to reach their objectives.

Even if the six contributions that comprise this Part do not pretend to completely cover the several challenges above discussed, nevertheless they provide a valid picture of the trends and solutions that may characterize future-generation network monitoring infrastructures and approaches. Chapter 19 presents the challenges and solutions for building extensible, programmable and high performance network monitoring solutions inspecting application layers. Especially programmability (which is also closely considered and advocated in Chapter 22) appears to be a key requirement to permit future monitoring infrastructures to rapidly and flexibly accommodate the continuously evolving needs posed by the emergence of new threats and new application-layer monitoring requirements. Chapter 20 clearly exemplifies how monitoring tasks are no more confined to the network layer, by showing techniques for detecting frauds and misuse of telephone services and combat spam over Internet telephony. Chapter 21 uses application layer semantics for monitoring and reducing overload situations in IP-based telephone networks using the SIP protocol. Chapter 22 brings about, again, the need for programmability from a different perspective, i.e., by specifically targeting the design issues and challenges behind programmable monitoring probes, and the possibility to support processing and filtering means directly on the probe itself, thus moving from a traditional centralized vision to a distributed and highly scalable modern "in-network" monitoring vision, where monitoring tasks are directly supported inside the network itself. Chapter 23 can be considered as a concrete example of such an in-network processing and filtering vision, as it discusses the rationale for offloading central intrusion detection systems by

implementing approximate intrusion detection rules directly on probes, and presents a proof-of-concept hardware implementation of a probe capable of supporting rules from the topmost known SNORT intrusion detection system. Finally, Chapter 24 addresses the very important and timely issue of how to protect network customers' privacy without compromising information usability for monitoring purposes. This chapter specifically investigates the fundamental principles and requirements for a privacy-aware ontological model in the semantic domain of monitoring-data management and exchange. It proposes a rule-based approach for specifying and automatically enforcing the appropriate privacy policies, and advocates a clean separation between data models and security semantics.

Part VI analyzes the issues and tradeoffs of bringing trustworthiness to digital content and its distribution. The Internet was originally conceived as an "Internet of Hosts", whose underlying protocols were designed to support exchange of simple unstructured information between well-identified nodes. Today, by contrast, it is becoming an Internet of Things (devices and appliances associated with their own IP address), an Internet of Services (in which users in different localities access different functionalities on different hosts), an Internet of Media (shared and managed across different networks) and an Internet of People (boosted by the explosion of social networking and the emergence of the Web 2.0 paradigm). In these "new Internets", the key elements are no longer "hosts" but data and services (or content). As one author put it, "People value the Internet for what content it contains, but communication is still in terms of *where*". In other words, what we are observing is a shift from "host-centric networking" to "content-centric" or "data-centric" networking.

This shift imposes new requirements on middleware, on the underlying networking functionality and on the way content is codified, formatted, described and exchanged in the network. Regarding the organization of content, several of these needs are addressed e.g. by existing MPEG standards. For instance MPEG-21 already defines standard ways of providing meta-information and standard ways of describing the content and structure of complex "Digital Items". However, there is the need of extending the ability to manage and trade "classical" media (e.g. video, music) digital objects to a broader range of digital objects, including descriptors for Real World Objects (RWO), services and people, meeting at the same time new requirements coming from such extended environment. Regarding middleware, there is the need of providing APIs to dynamically define and encapsulate new classes of content, and related meta-information, to create packages of different classes of information resource, to guarantee their security and privacy and integrity, to name them, to support semantic interpretation of metadata and tags, to search for them, filter them, read and write their attributes and content, adapt them for use on different machines, copy them, test their validity and efficiently synchronize them across multiple machines. The MPEG-M emerging standard, an extension of the former MXM MPEG platform, is addressing a significant part of these issues. As regards networking functionality, content-centric architectures are being proposed, where the network layer directly

provides users with contents, instead of providing communication channels between hosts.

Chapter 25 presents an outlook on the definition and implementation of distributed architectures that enable the development of distributed multi-media applications on top of them, while offering Digital Rights Management (DRM) features. Chapter 26 develops from the consideration that innovative networks must be aware of which content is actually transported and introduces Scalable Video Coding as an important tool for such networks. Finally, Chapter 27 identifies the main functionality of a content-centric network, discusses pros and cons of literature proposals for an innovative, content-centric network layer and draws conclusions stating some general requirements, which a content-centric network layer should satisfy.

Contents

Part I
New Visions for a Trustworthy Internet

Chapter 1
Publish–Subscribe Internetworking Security Aspects

Nikos Fotiou, Giannis F. Marias and George C. Polyzos

Abstract Publish–Subscribe is a paradigm that is recently receiving increasing attention by the research community, mainly due to its information oriented nature. Although the publish–subscribe paradigm yields significant security advantages over the traditional send–receive one, various security and privacy challenges are raised when it comes to the design of an internetworking architecture that is solely based on this paradigm, such as the Publish Subscribe Internet (Ψ) architecture. Ψ is the main outcome of the Publish–Subscribe Internet Routing Paradigm (PSIRP) project, which was launched with the ambition to develop and evaluate a clean-slate architecture for the future Internet based on the publish–subscribe paradigm. Availability, security, privacy and mobility support are considered as core properties for this new form of internetworking, instead of being provided as add-ons, as in the current Internet. This paper discusses the security and privacy properties of and challenges for publish–subscribe internetworking architectures and specific techniques and solutions developed in PSIRP for Ψ.

Keywords Clean slate · Future Internet · PSIRP · Publish/subscribe · Security

N. Fotiou (✉) · G. F. Marias · G. C. Polyzos
Mobile Multimedia Laboratory, Athens University of Economics and Business, Patision 76, 104 34 Athens, Greece
e-mail: fotiou@aueb.gr

G. F. Marias
e-mail: marias@aueb.gr

G. C. Polyzos
e-mail: polyzos@aueb.gr

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_1,

1.1 Introduction

The Publish–Subscribe paradigm has been in the spotlight of recent research efforts. Its information oriented nature, the decoupling it offers between information providers and information consumers as well as its location-identity split properties, have inspired a variety of—mainly overlay—architectures that focus on multicast [6], mobility [15], indirection [29] as well as on caching [16].

Publish–Subscribe architectures are composed of three main components; publishers, subscribers and a network of brokers [8]. Publishers are information providers that 'publish' information (advertisements). Subscribers on the other hand are information consumers that express their interest in specific pieces of information by issuing subscriptions. Brokers are responsible for matching publications with subscriptions and initiate the (information) forwarding process from information providers towards information consumers. The broker, responsible for the publication–subscription matching, is often referenced to as the rendezvous point and, therefore, the network of brokers is usually referred to as the rendezvous network. Publication and subscription operations are decoupled in time and space allowing for the support of mobility as well as anonymization mechanisms. Moreover a publication can be provided by multiple nodes and similar subscriptions can be aggregated, creating opportunities for multicasting and multihoming. Inherently, the publish–subscribe paradigm has many security advantages compared to the commonly used end-to-end, send–receive oriented paradigm.

PSIRP (Publish–Subscribe Internet Routing Paradigm),[1] an EU FP7 funded research effort, has designed, implemented in prototypes, and initially evaluated a clean-slate, information oriented future Internet architecture; we call it the *Publish–Subscribe Internet* (PSI) architecture, Ψ for short. This architecture aims at overcoming most limitations of the current Internet and at emphasizing the role of information as the main building block of the (future) Internet. This new architecture is based on a paradigm completely different from the current one. Ψ is based on pure, through-the-stack application of the Publish–Subscribe paradigm. Moreover by abiding to the Trust-to-Trust (T2T) principle [4], i.e., all functions take place only in trusted points, the Ψ architecture considers security as a building block of its architecture rather than as an 'add-on'. Ψ harvests the security advantages the publish–subscribe paradigm offers, whilst Ψ-specific security mechanisms are also incorporated.

The purpose of this paper is twofold: to give an overview of the security features of and challenges for the publish–subscribe paradigm, as well as to show the additional techniques and mechanisms developed in PSIRP in order to secure the Ψ architecture. The remainder of this paper is organized as follows. Section 1.2 highlights some of the security and privacy challenges that exist in publish–subscribe architectures. Section 1.3 presents the security advantages of the publish–subscribe paradigm. Section 1.4 overviews the Ψ architecture and its specific

[1] http://www.psirp.org

security solutions. Section 1.5 investigates how other, related architectures handle security requirements. Finally, our conclusions as well as ideas for future work are presented in Sect. 1.6.

1.2 Security and Privacy Challenges in Publish–Subscribe Architectures

As previously mentioned, in the publish–subscribe model, producers publish event notifications to announce information availability and consumers subscribe to specific information items to explicitly declare their interest. Matching is achieved through the rendezvous network, which is envisioned as a distributed service that spans over a large number of providers and administrative domains. In the case where one or more matches are provided by the rendezvous service, then a particular sub-graph over the network topology is determined and activated to support a multihomed and multicasted communication service that transports information elements from publisher(s) to subscriber(s).

Security issues and requirements that arise in a global-scale publish–subscribe system have already been extensively addressed. Wang et al. [31] as well as Lagutin et al. [21] have specified security requirements for a publish–subscribe architecture, whereas Wun et al. [33] have identified and classified possible DoS attacks in content-based publish–subscribe systems. Various mechanisms have been developed in order to secure publish–subscribe systems—such as Eventguard [28]—and most of them base their operation on traditional security mechanisms, adapted to the concept of the publish–subscribe paradigm. In this paper we are focusing on security, trust, and privacy requirements focusing on a different level of abstraction and trying to enrich the existing work with recent results for the publish–subscribe paradigm.

In the information level, integrity, authenticity and validity of information are required. Integrity protection methods will ensure that any violation or fabrication of information elements' content will be detectable. Authenticity means that the information that is received by the subscriber is identical with the subscriber's initial request, and it is not forged. Validity means that the information items announced by the publisher, matched with the subscriber's request, and then forwarded to the subscriber are identical. Detecting integrity violation is a task that mainly is based on public key certificates and signatures, and, thus, it requires trusted third parties or bilateral trust (e.g., symmetric secrets, or HMAC key-based approaches). On the other hand, publication and subscription operations might be decoupled in time. Thus, subscribers might never recognize the publishers' identities, or even their certificates. Thus, information integrity verification should be assisted by the rendezvous-network. In order to avoid bottlenecks due to processing or signing every information element, rendezvous nodes might produce sequences of integrity evidences, such as TESLA seeds if a TESLA approach [25] has been

adopted between publication end-points and consumers. Verifying authenticity and validity of the information requires a different, reaction-oriented approach, which is based on subscriber's evaluation on the received information. Such an approach will rank published information elements, and recommend the accurate ones, avoiding DoS attacks [11] or spamming [12].

At the application layer, a main security challenge is the design of a mechanism that grants to subscribers the appropriate access privileges to publication announcements. This is akin to making confidential the existence, and not the content, of publications. Assuming that publishers are always privileged to submit events and announcements, the rendezvous network should enforce an access framework that makes the notifications reachable to preferred subsets of subscribers. For application-level access control, such subsets are formed using scopes [9], role-based access control [3, 26], as well as identification and authentication schemes [24]. On the other hand, publication content confidentiality is achieved mainly through encryption. Finally, when a forwarding topology will be deployed to transport information to subscribers, then there is a potentially strong anonymity requirement to unlink the information and the publisher and subscribers among themselves and from the networking attachment and relay points.

From the subscriber's privacy point of view, a central objective is to unlink his identity from his subscription interests, e.g., by supporting anonymous subscriptions. Subscription privacy might rely on an anonymity framework related to trusted proxies (anonymizers) that receive and process the original request, change its time reference, hide the subscriber's identity and obfuscate his network attachment point. This approach might introduce significant delays, but fulfills the demand for strong anonymity support at the network layer. Additionally, such a system should be designed and deployed appropriately to avoid attacks that have been reported on mix-based privacy enhancement approaches, such as traffic analysis, blending and trickle attacks [32].

1.3 Publish–Subscribe Security Features

The publish–subscribe paradigm can be seen as a remedy to the imbalance of power between senders and receivers in the traditional send–receive paradigm. With the original Internet architecture, the network will make a best effort attempt to deliver whatever any sender sends, irrespective of the interest of and no matter the cost for the receiver and the network(s). This imbalance is often accused for the increasing number of (Distributed) Denial of Service (DDoS) attacks, as well as for the emergence of spamming. In publish–subscribe systems there is no information flow as long as the receiver has not expressed interest on a particular piece of information, i.e, the receiver in a publish–subscribe architecture is able to instruct the network which pieces of information shall be delivered to it. Moreover, and even though the model is so powerful so that there can be subscriptions before the corresponding publications have been published, no information is requested

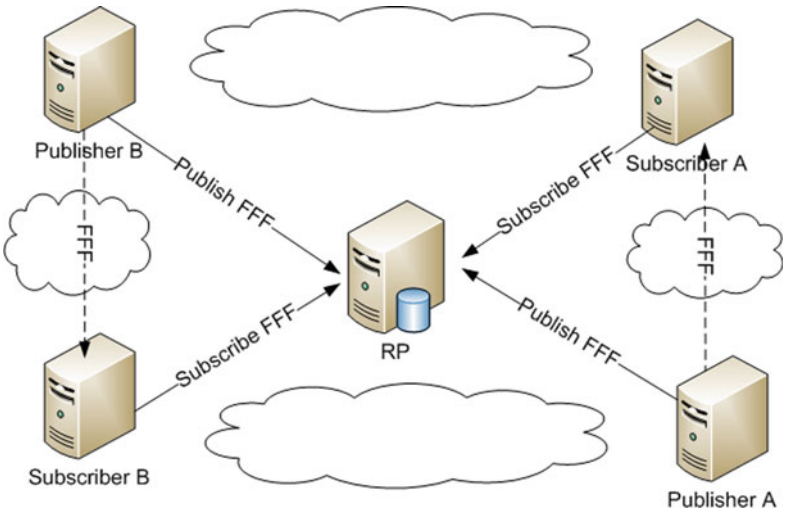


Fig. 1.1 Example of multihoming in a publish–subscribe architecture

from a publisher, unless the publisher has explicitly denoted the availability of that information, i.e., not before the publisher has issued a publication message (for this particular piece of information).

Publication and subscription operations are decoupled in time and space, i.e., they do not have to be synchronized neither do they block each other. Moreover publishers and subscribers do not communicate directly and they can hide their identity as—in general—subscribers are only interested for the information itself rather than on who provides it, and publishers—usually—disseminate publications using multicast so they cannot (and usually should not) be fully aware of the publication's recipients. Therefore, anonymity can be easily achieved in publish–subscribe architectures. Moreover having a point in the network where subscription and publications are matched, effective deployment of access control mechanisms is enabled.

Publish–Subscribe architectures offer great availability. The rendezvous network of a publish–subscribe architecture is usually implemented using a DHT. DHTs provide significant load balancing—usually at the cost of some communication stretch. Moreover in a publish–subscribe architecture multihoming can be easily achieved, as multiple publishers may advertise the same publication to a Rendezvous Point (RP), therefore a RP has a number of options with which it can satisfy a subscription. Figure 1.1 shows an example of multihoming in a publish–subscribe architecture. Publishers A and B, both publish publication FFF. Subscribers A and B subscribe to this publication. For each subscription message the RP knows two publishers that can provide the publication matched, therefore for each subscription message it could choose the publisher that is closer (in any sense) to the respective subscriber, e.g. here, it chooses publisher A to serve subscriber A and publisher B to serve subscriber B.

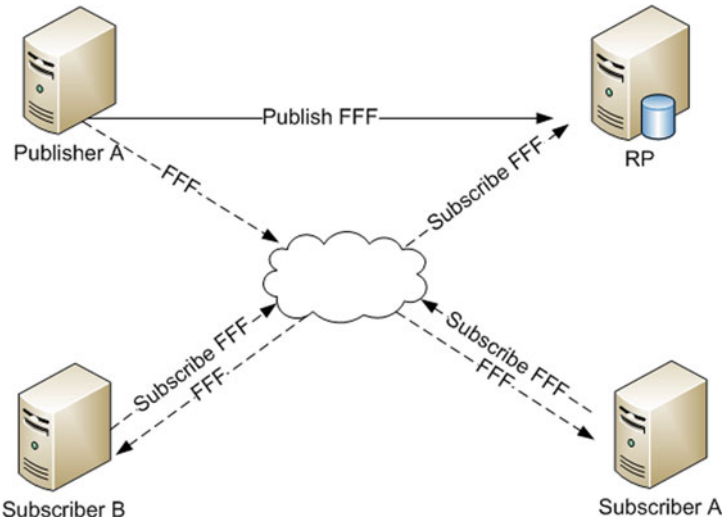


Fig. 1.2 Resource sharing in a publish–subscribe architecture using subscriptions aggregation and multicast

Publish–subscribe architectures allow for subscription aggregation and they create opportunities for multicast to be useful, therefore in these architectures resource sharing can be achieved, leading to greater availability. In Fig. 1.2 both subscribers A and B subscribe to publication FFF. The subscription messages are aggregated within the networks, when following the same path towards the RP. Moreover publisher A forwards a single data flow, which is copied (bifurcated) in an appropriate place in the network in order to serve both subscribers.

1.4 The Ψ Architecture

The core element of the Ψ architecture is information; information is everything and everything is information [30]. In Ψ every piece of information is identified by a unique, flat, self-certified identifier, known as the *rendezvous identifier* (RId). Information is organized in *scopes*. Scopes are physical or logical structures that facilitate the finding as well as access control over a piece or collection of information. A physical scope can be for example a corporate network, whereas a logical scope can be a group of friends in a social network. Scopes can be included within each other, creating a flexible structure. Scopes are identified by a flat identifier known as the *scope identifier* (SId). Each SId is managed by a rendezvous point (RP) which can be a single *rendezvous node* or a large *rendezvous network*.

The publication operation in Ψ involves three steps [10]; initially the SId of the publication scope is identified, then the RId of the publication is created and,

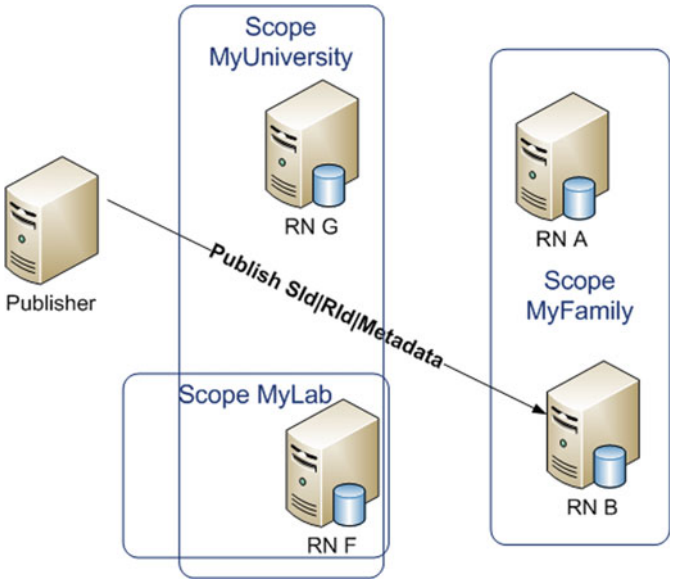


Fig. 1.3 Publication in a Ψ network

finally, the publication is published in i.e. the publication message, including the Rid and Sid, is sent to the RP responsible for handling this SId. The publication message may also contain *metadata*—such as size of the data, encoding and other general information about this publication. Figure 1.3 shows the publication operation in a Ψ network with three scopes; the scope MyUniversity and its sub-scope MyLab and the scope MyFamily. As it can be seen in this figure, a publisher issues a publication to the scope MyFamily. The publication message should contain a scope-unique publication identifier (RId), the MyFamily scope identifier (SId) as well as metadata that describe this publication. The publication message reaches the rendezvous node RN B, which is part of the MyFamily rendezvous network.

The subscription operation involves the identification of the SId and RId of a publication—which can be done, for instance, with the help of a search engine—and the sending of a subscription message. Initially the subscription message will be forwarded to the appropriate scope as all the other scopes are not aware of the publication in question. When the subscription reaches the appropriate scope it will be forwarded to the publication RP. The network is responsible for routing publication and subscription messages towards the RP as well as for forwarding publications from publishers towards subscribers. Figure 1.4 shows the subscription operation. A subscriber subscribes to an already published publication. When the subscription message reaches the appropriate RP, and as long as there is a publication that matches this subscription message, the RP creates a forwarding path, from the publisher towards the subscriber, and instructs the publisher to send

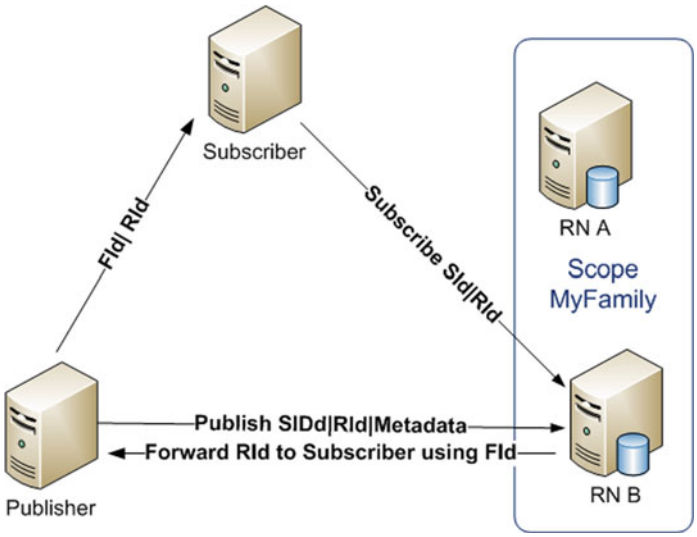


Fig. 1.4 Ψ subscription: initially (e.g.) the publisher issues a publication, then a subscriber, subscribes to this publication and the rendezvous point instructs the publisher to forward this publication to the subscriber

the publication using a specifically created identifier (FId) for this path. A forwarding path is realized through zFilters [14], a Bloom filter based structure that contains the link identifiers that a data packet must traverse in order to reach its destination(s). Ψ uses a slow path for signaling, i.e., publication and subscription messages, and a fast path for data forwarding. Moreover multicast is the preferred delivery method.

1.4.1 Ψ-Specific Security Mechanisms

Security in Ψ plays an important role and trust is at the center of a Ψ declared principle. Security mechanisms are considered at all levels of the architecture. Information in Ψ is transmitted in encrypted packets using the Packet Level Authentication (PLA) technique [19]. PLA is a novel mechanism, applied in Ψ, for protecting the network based on the assumption that per packet public key cryptographic operations are possible at wire speed in high speed networks with the help of new cryptographic algorithms and advances in semiconductor technology. Moreover when applied in wireless environments PLA has been proven to offer significant energy efficiency [20].

As already described Ψ's forwarding mechanism is based on the formation of a Bloom filter—called zFilter—that describes the path that a data packet should follow [14]. The computation of the zFilter is based on the identifiers of the links

that compose the data path. These identifiers are dynamically generated every time a zFilter is created, making this way almost impossible for an attacker to create crafted zFilters or link identifiers that will lead to DoS attacks or to information leakage. Forwarding using zFilters is achieved at line speed, leading to excellent performance and scalability. Network attachment in Ψ [17] assures proper user authentication protecting both users from improper configuration as well as the network from (D)DoS attacks that can be caused by malicious users who repeatedly try to attach themselves to a Ψ network.

At the higher layers of the architecture, existing security mechanisms can be used. Nikander and Marias [22] studied the application of existing work on cryptographic protocol analysis in a pure publish–subscribe architecture and found out that, even if networking protocols are revised very drastically, current cryptographic protocol analysis can be applied to a certain extent, with only minor modifications, mostly on the notation side. Moreover, novel trust mechanisms should be considered applied to information ranking [12] rather than ranking end-users.

Ψ security is going to be primarily based on the notion of scopes. Although not yet fully designed and implemented, scopes are expected to control information dissemination as well as to play a significant role in applying access control policies, as well as accounting mechanisms. Scopes are expected to be Ψ's information firewalls.

1.5 Security Aspects of Comparable Internetworking Architectures

CCNx [7] (Content-Centric Networking, now termed Named Data Networking: NDN) is an ongoing research project that investigates the potential of an information-oriented Internet architecture. In contrast to Ψ, CCNx proposes an architecture organized using hierarchical naming [13]. Moreover CCNx uses a broadcast-based mechanism for information location, rather than a rendezvous driven one. CCNx does not rely on flat self-certified identities, it rather uses a scheme that assures the relationship between publications and their identities and it provides validity, provenance, and relevance [27]. In this scheme every publisher is allowed to generate a user-friendly tag label for their publication, which in a next step is incorporated into the body of the publication as a digital signature. This digital signature is generated by applying the publisher's public key over the publication's data and the publication label. When a subscriber receives the publication, and provided that the publisher is reliable, he is able to verify that the publication he received matches its label. On the other hand in case of a malicious publisher that uses forged labels, this publisher can be held accountable for his behavior, as its public key has been used in order to generate the publication's digital signature.

The Data-Oriented Network Architecture (DONA) [18] and Routing on Flat Labels (ROFL) [5] are two pioneering architectures that introduced flat identifiers. DONA aims at replacing DNS with flat self-identifying labels that will enable data location and retrieval. In contrast to Ψ, DONA uses the same path, for information location and forwarding. DONA's main security mechanism is its self-certified naming. DONA names are organized around principals and they are of the form P:L, where P is the cryptographic hash of the principal's public key and L is a label chosen by the principal, who ensures that these names are unique. Every publication is accompanied by a metadata file that includes the principal's public key as well as her digital signature over the publication data. Users in DONA are expected to learn a publications' name using external, reliable mechanisms. In order to defend against DoS attacks, DONA relies on IP-level mechanisms, as well as on the limits that providers will pose on users' publications and subscriptions. Finally DONA assumes the existence of third trusted parties for public key status retrieval and revocation.

ROFL creates an internetworking architecture in which routing takes place solely based on the data—flat—identifiers. In ROFL there is no information hierarchy, as there is in Ψ (with the usage of scopes) and DONA. ROFL security is also based on self certified identities. In ROFL, in every network node, i.e., router or host, a unique ID is assigned, which is tied to a public–private key pair. This key pair is used to sign-verify every packet that traverses the system. ROFL secures its routing infrastructure by using the so-called *filtering* and *capabilities* techniques. With *filtering*, every host can control its reachability and therefore filter out malicious hosts. With *capabilities* the architecture is able to perform fine-grained access control. Whenever a (legitimate) host requests the creation of a network path, a *capability* token is provided, which proves that the host has the proper access control credentials for this path. *Capability* is a cryptographic token designating that a particular source (with its own unique object identifier) is allowed to contact the destination.

The Internet Indirection Infrastructure (i3) [29] and the Host Identity Protocol (HIP) [2] are two rendezvous-based overlay solutions that aim at supporting mobility, multicast and multihoming. Ψ's rendezvous and topology processes use similar concepts, at all levels of the architecture.

i3 implements an IP overlay network that replaces the point-to-point communication model with a rendezvous-based paradigm. In i3 *sources* (akin to Ψ publishers) send packets to a logical identifier, whereas *receivers* (akin to Ψ subscribers) express interest in packets by inserting a trigger into the network. A distributed lookup service is responsible for matching triggers with packets and an overlay network of i3 nodes is responsible for forwarding packets. An i3's extension, known as the *Secure-i*3 [1], further enhances the security of the proposed architecture by allowing hosts to hide their IP address as well as to defend against DoS attacks without introducing new vulnerabilities. IP address hiding is accomplished with the usage of the so-called *private IDs*; when an end-host issues a new trigger, instead of using its real IP address, it uses the public ID of an i3 (reliable) node that acts as the end-host's representative. The public ID of this i3

node is the private ID of the end-host. Even if the representative node removes its public ID it will not affect the already established end-host's connections. Every node in i3 may have multiple public IDs. In case of DoS attacks a node may remove all of its public IDs to eliminate the attack, or remove some of them in order to mitigate the attack. Moreover, puzzles can be used as a countermeasure against DoS attacks; before a suspicious host is allowed to send a new packet, it is requested to solve a cryptographic puzzle. Finally, hosts in i3 can manipulate the path that a packet should follow in order to reach them, this way they are able to circumvent parts of the network that are under attack.

HIP introduces a new layer that decouples host identity from location identity in the internetwork stack, between the IP layer and the transport layer. When HIP is used, the applications no longer connect to IP addresses, but to separate *Host Identifiers*. A Host Identifier is a cryptographic hash of the host's public key, which in turn, is used for securing communication between hosts. The resolution from a Host Identifier to an IP address can be achieved either by using a DNS-like mechanisms or a DHT. *Host Identity Indirection Infrastructure* (Hi3) [23] is the secured version of the HIP protocol, which utilizes Secure-i3's rendezvous principles. Secure-i3 is used in order to perform Host Identifier to IP address resolution, whereas IPSec is used for the rest of the communication between hosts.

1.6 Conclusions and Future Work

The Publish–Subscribe paradigm achieves a significant shift from the current end-host driven internetworking towards an information oriented Internet architecture. This paradigm offers significant security advantages, including greater availability and enhanced privacy. The opportunities for multicast, mobility support and caching, as well as, the decoupling it offers between the communicating parties, make the publish–subscribe paradigm a strong candidate for a future internetworking architecture. Nevertheless various security and privacy challenges remain and further research is needed in order to identify and tackle them. Towards this direction the PSIRP project has created the so-called Ψ architecture; a clean slate Internet architecture that is based on the publish–subscribe paradigm. The Ψ architecture demonstrates the significant capabilities of this paradigm and through the development of Ψ-specific security mechanisms shows the road towards a secure future internetworking architecture.

The research in this field is a very active ongoing effort. Various research projects around the world investigate the potential of new internetworking architectures based on the publish–subscribe paradigm—or other similar ones. Security remains in the spotlight of all these research efforts. As far as the Ψ architecture is concerned, its research and development continues during the EU FP7 PURSUIT[2]

[2] http://www.fp7-pursuit.eu/

project, which plans to further explore security, privacy and trust issues of this architecture, as well as to create novel mechanisms and evaluate them including aspects of them experimentally over the newly established Ψ testbed which spans Europe.

Acknowledgements The work reported in this paper was supported in part by the FP7 ICT project PSIRP, under contract ICT-2007- 216173.

References

1. Adkins, D., Lakshminarayanan, K., Perrig, A., Stoica, I.: Towards a more functional and secure network infrastructure. Tech. Rep. UCB/CSD-03- 1242, EECS Department, University of California, Berkeley (2003). http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/6241.html
2. Al-Shraideh, F.: Host identity protocol. In: Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on, pp. 203–203 (2006)
3. Belokosztolszki, A., Eyers, D.M., Pietzuch, P.R., Bacon, J., Moody, K.: Role-based access control for publish/subscribe middleware architectures. In: DEBS'03: Proceedings of the 2nd International Workshop on Distributed Event-Based systems, pp. 1–8. ACM, New York (2003)
4. Blumenthal, M.S., Clark, D.D.: Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. ACM Trans. Internet Technol. **1**(1), 70–109 (2001)
5. Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K., Stoica, I.: Rofl: routing on flat labels. In: SIGCOMM'06: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 363–374. ACM, New York (2006)
6. Castro, M., Druschel, P., Kermarrec, A.M., Rowstron, A.: Scribe: a large-scale and decentralized application-level multicast infrastructure. Sel. Areas Commun. IEEE J. **20**(8), 1489–1499 (2002)
7. CCNx: http://www.ccnx.org (2010)
8. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
9. Fiege, L., Zeidler, A., Buchmann, A., Kilian-Kehr, R., Mühl, G.: Security aspects in publish/subscribe systems. In: Proceedings of Third International Workshop on Distributed Event-based Systems (DEBS04 (2004)
10. Fotiou, N., Polyzos, G.C., Trossen, D.: Illustrating a publish–subscribe internet architecture. In: Proceedings of the 2nd Euro-NF Workshop on Future Internet Architectures. Santander, Spain (2009)
11. Fotiou, N., Marias, G., Polyzos, G.C.: Fighting spam in publish/subscribe networks using information ranking. In: Proceedings of the 6th Euro-NF Conference on Next Generation Internet Networks (NGI). Paris, France (2010)
12. Fotiou, N., Marias, G., Polyzos, G.C.: Information ranking in content-centric networks. In: Proceedings of the Future Network and MobileSummit 2010. Florence, Italy (2010)
13. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: CoNEXT '09: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, pp. 1–12. ACM, New York (2009)
14. Jokela, P., Zahemszky, A., Esteve Rothenberg, C., Arianfar, S., Nikander, P.: Lipsin: line speed publish/subscribe inter-networking. In: SIGCOMM '09: Proceedings of the ACM

SIGCOMM 2009 Conference on Data Communication, pp. 195–206. ACM, New York (2009)
15. Katsaros, K., Fotiou, N., Polyzos, G., Xylomenos, G.: Overlay multicast assisted mobility for future publish/subscribe networks. In: Proceedings of the ICT Mobile Summit, Santander, Spain (2009)
16. Katsaros, K., Xylomenos, G., Polyzos, G.C.: A hybrid overlay multicast and caching scheme for information-centric networking. In: Proceedings of the 13th IEEE Global Internet Symposium. San Diego, CA, USA (2010)
17. Kjallman, J.: Attachment to a Native Publish/Subscribe Network. In: ICC Workshop on the Network of the Future. Dresden, Germany (2009)
18. Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. In: SIGCOMM '07: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 181–192. ACM, New York (2007)
19. Lagutin, D.: Redesigning internet-the packet level authentication architecture. Licentiates Thesis in Computer Science, Helsinki University of Technology, Espoo, Finland (2008)
20. Lagutin, D., Tarkoma, S.: Public key signatures and lightweight security solutions in a wireless environment. Smart Spaces Next Gener Wired/Wireless Netw **5764**, 253–265 (2009)
21. Lagutin, D., Visala, K., Zahemszky, A., Burbridge, T., Marias, G.: Roles and security in a publish/subscribe network architecture. In: Proceedings of the 2010 IEEE Symposium on Computers and Communications (2010)
22. Nikander, P., Marias, G.: Towards understanding pure publish/subscribe cryptographic protocols. In: Sixteenth International Workshop on Security Protocols, Cambridge, England (2008)
23. Nikander, P., Arkko, J., Ohlman, B.: Host identity indirection infrastructure (Hi3). In: Proceedings of the Second Swedish National Computer Networking Workshop SNCNW, Karlstad, Sweden (2004)
24. Pallickara, S., Pierce, M., Gadgil, H., Fox, G., Yan, Y., Huang, Y.: A framework for secure end-to-end delivery of messages in publish/subscribe systems, Proceedings of the 7th IEEE/ACM International Conference on Grid Computing (GRID 06), 215–222 (2006)
25. Perrig, A., Canetti, R., Tygar, J., Song, D.: The TESLA broadcast authentication protocol. RSA CryptoBytes **5**(2), 2–13 (2002)
26. Pesonen, L.I.W., Eyers, D.M., Bacon, J.: Encryption-enforced access control in dynamic multi-domain publish/subscribe networks. In: DEBS '07: Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems, pp. 104–115. ACM, New York (2007)
27. Smetters, D., Jacobson, V.: Securing Network Content. Tech. Rep., PARC (2009)
28. Srivatsa, M., Liu, L.: Securing publish–subscribe overlay services with eventguard. In: CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security, pp. 289–298. ACM, New York (2005)
29. Stoica, I., Adkins, D., Zhuang, S., Shenker, S., Surana, S.: Internet indirection infrastructure. IEEE/ACM Trans. Netw. **12**(2), 205–218 (2004)
30. Tarkoma, S. (ed.): PSIRP Deliverable 2.3, Architecture Definition, Component Descriptions, and Requirements (d2.3) (2010). http://www.psirp.org/
31. Wang, C., Carzaniga, A., Evans, D., Wolf, A.: Security issues and requirements for Internet-scale publish–subscribe systems. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences, pp. 3940–3947 (2002)
32. Wright, J., Stepney, S., Clark, J., Jacob, J.: Formalizing anonymity: a review. Report YCS 389. University of York, Department of Computer Science (2005)
33. Wun, A., Cheung, A., Jacobsen, H.A.: A taxonomy for denial of service attacks in content-based publish/subscribe systems. In: DEBS '07: Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems, pp. 116–127. ACM, New York (2007)

Chapter 2
Security Issues in Programmable Routers for Future Internet

Raul Cafini, Walter Cerroni, Carla Raffaelli and Michele Savi

Abstract Network resource virtualization and dynamic service configuration are key functionalities in future Internet scenarios and can take advantage of the network programmability paradigm. As a key system thought for next-generation networks, a programmable router is a modular node to support this context. It relies on high-performance optical switching fabric and modular software organization, to seamlessly meet the dynamic requirements of emerging and future applications. In this chapter, the security aspects arising when customers and service providers access node and service programmability are discussed. Secure service support based on control and forwarding element separation is proposed and tested in a software router context to show the feasibility and the effectiveness of the approach.

Keywords Future Internet · Network Programmability · Multi-service Router Architecture · Security in Programmable Routers · Software Router

R. Cafini · W. Cerroni (✉) · C. Raffaelli · M. Savi
DEIS - University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: walter.cerroni@unibo.it

R. Cafini
e-mail: raul.cafini@unibo.it

C. Raffaelli
e-mail: carla.raffaelli@unibo.it

M. Savi
e-mail: michele.savi@unibo.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_2,

2.1 Introduction

The traditional *hourglass* model, where a simple yet powerful protocol such as IP acts as the globally unifying element whereas most of the complexity is left to the communication endpoints, has been one of the main reasons for the explosive growth of the Internet in the last two decades. On the other hand, the very same model is showing its limitations in dealing with the evolution of today's networked services that are pervading many aspects of our lives. Several features are required by state-of-the-art applications and services, e.g., flexibility, quality of service, mobility, wired–wireless convergence, multi-domain integration, reliability, security, limited impact on the environment, to name a few. Today these aspects are either neglected or addressed adopting single and specific solutions that are not part of a holistic approach to the problem.

From another point of view, the current network-centric approach is capable of offering mainly technology-dependent transport paradigms, semi-static connectivity services and limited knowledge of real user and service needs. Therefore, a completely different architecture must be deployed for what will become the Internet of the Future, moving from a network-centric approach to a user-, service- and content-centric philosophy [1]. Under these assumptions, the network layer of the Future Internet needs to be re-engineered with increased level of flexibility and dynamism, implementing built-in network functionalities capable of offering on-demand virtual communication resources based on high-level service needs.

Node and network programmability represents a key concept to achieve this ambitious goal, considering in particular the potentials of a network where connected nodes are not simply used to forward information from source to destination according to a fixed paradigm, but they play an active part within an integrated system that manages communication resources in a highly flexible way. Programmable nodes are capable of providing open and accessible interfaces for controlling switching and routing facilities and other network-layer functions on top of a common physical infrastructure, which can be dynamically reconfigured at several levels by *virtual operators* to offer different on-demand connectivity services to their customers.

Among the major actors involved in the definition of the Future Internet infrastructure, an important role is certainly played by optical networks based on emerging photonic switching technologies, thanks to their advantages in terms of huge bandwidth, reduced power consumption and small footprint, to name a few [2]. A programmable approach to the deployment of such networks allows to take advantage of all the benefits they offer while adopting a scalable and cost-effective resource sharing solution. Obviously, as with any other shared resource management scheme, security and reliability issues come into play and must be properly addressed.

The main contribution of this chapter is the analysis of node programmability issues from a security perspective, assuming the multi-service oriented node architecture proposed in Sect. 2.3 as a use case. Authentication, authorization,

integrity, confidentiality, protection and availability aspects are discussed in detail and possible solutions based on well-established techniques are sketched. A software router testbed of both control and data planes in a programmable node is presented to emulate secure and flexible service provisioning and to show the feasibility and the effectiveness of the proposed approach through sample tests.

The chapter is organized as follows. Section 2.2 discusses the concept of network programmability, including a historical perspective as well as current standardization and research trends. Section 2.3 introduces the architecture of the proposed multi-service programmable router, whereas Sect. 2.4 discusses its main security aspects. Then Sect. 2.5 illustrates how security can be enforced in the control plane adopting current standard techniques. Finally, Sect. 2.6 provides experimental validation in a software router context and Sect. 2.7 concludes the chapter.

2.2 Network Programmability: Past, Present and Future

To understand the potential implications and advantages of the network programmability concept, let us consider a possible use case in the near future where a given user needs to backup 100 gigabytes of data to a remote storage location using her/his broadband access connection. While this is unpracticable with current ADSL uplink bit rates, the time required to transfer such an amount of data with advanced access technologies available today, such as 100 Mpbs FTTH connections, is more than two hours. However, the user would be very happy to have this backup operation completed in much less time, e.g. in the order of a couple of minutes, which would require a 10 Gbps access bandwidth (assumed to be feasible in the next 5–10 years).

On the other hand, for the rest of the day the user does not need more than a few hundreds of Mbps for her/his normal network activities. Therefore, having the 10 Gbps bandwidth guaranteed only during the backup transfer by means of a user-controlled, dynamic connectivity service would provide many benefits for both the user, in terms of savings on the cost of the service, and the network operator, in terms of fast service provisioning and efficient resource management. This implies a vision of the network as a programmable system, where the user is capable of setting up specific network layer functions at will, similarly to what happens when programming general-purpose computational functions in computers.

Many other examples can be made to show the advantages of programmable networks, but at this point a general definition is mandatory. Borrowing from [3], the concept of *network programmability* can be defined as *the main attribute of an open architecture for network devices allowing third party code to be executed in order to apply changes in network functionality, or in the way the network is operated, in both the control plane and the data plane*.

The first wave of research activities on network programmability took place in the late 1990s with two main approaches, namely *programmable networks* and

active networks [4]. The former is an open approach where standardized programming interfaces allow customer applications to activate and manage services by reconfiguring low-level routing and switching resources, e.g., to create dynamic virtual private networks at different levels of granularity [5] or to deploy suitable transport protocols based on Quality of Service (QoS) application needs [6].

Active networks adopt a more radical approach by assuming that each packet may carry not only input data (e.g. the IP header) to the routing and networking functions, but also small programs to be executed on the nodes to customize those functions [7, 8]. This can be done in two ways. On one hand, a traditional out-of-band management scheme can be implemented where a few "special" packets are used to program the node, while ordinary packets are simply forwarded using the newly programmed functions [9]. On the other hand, in-band management can be achieved when every packet carries a program fragment to be executed on intermediate active nodes to change their current state [10].

Recently, network programmability issues gained renewed interest from the research community due to the advances in several related topics, including among others: virtualization technologies, programmable network hardware, software router implementations, overlay network architectures [11]. One of the main challenges for network operators and service providers is *the difficulty to rapidly and safely deploy new services and service features* [3]. Network programmability is envisioned as one of the key supporting factors to deal with this challenge.

Meanwhile, some standardization initiatives are taking place that can actively contribute to the actual deployment of programmable networks. For instance, the IETF defined a router architecture framework named *ForCES* where the control functions are kept separated from the forwarding functions [12]. A standard protocol is also available to replace proprietary communications among control and forwarding elements [13], so that network boxes can be turned into multi-vendor equipment where control and forwarding subsystems can be developed and can evolve independently. This approach clearly goes in the direction of simplifying the implementation of router programmability, as testified by some recent works [14, 15].

On another side, IETF issued a standard protocol for network configuration called *Netconf* aimed at providing a unified, cross-vendor, interoperable management interface for automated network device configuration [16]. The Remote Procedure Call (RPC) paradigm and XML data format adopted by Netconf allow heterogeneous network devices to be controlled by means of standard application programming interfaces, making this protocol a powerful tool for implementing the signaling scheme required by a programmable network.

Considering now a future perspective, next generation networking scenarios will be characterized by strong virtualization and dynamic reconfiguration of network resources. In particular, the nodes will not only provide very high bandwidth and aggregate capacity, but will also be required to be flexible enough to easily map the requests coming from the network control plane onto the available switching and routing resources, according to their quality of service needs and traffic characteristics [17, 18]. In addition, smart network control

functions will be required to provide fast, semantically rich and flexible signaling procedures [19]. Therefore, it is reasonable to assume that network programmability will be an integral part of the Future Internet deployment. Some experiments have been recently carried out on high-capacity router modules [20], showing also that integration between optical data plane and application-aware control plane is feasible [21]. The idea is to add an adequate level of flexibility to high-capacity switching matrices, typically exploiting emerging optical technology at different levels of granularity.

In this perspective, a *programmable optical node* is such that different and multi-granular switching and transport paradigms can be simultaneously managed over a high-capacity, possibly photonic, switching matrix. This way, a scalable and cost-effective solution can be available to service providers, as they can support different transport services over the same physical node and network in a highly flexible way. At the same time, the programmable node concept raises new issues and opportunities regarding security and reliability. In fact, being it thought as a shared, high-capacity system, it must be protected from unauthorized access to node resources and functionalities. At the same time, its programmable and modular characteristics can be exploited to effectively implement network security.

2.3 Multi-Service Programmable Router Architecture

A *multi-service programmable router* is defined here as a node supporting network functions which can be activated on-demand by simply adding or re-configuring specific software modules that control the underlying switching and transmission hardware. The goal is to implement a flexible architecture where an external Internet Service Provider (ISP), referred here on as *Guest Operator* (GO), is able to dynamically use part of the existing infrastructure managed by a *Host Operator* (HO) to offer different connectivity services to its own customers. To this end, the HO must be able to manage different network-based applications according to their specific requirements and translate technology-specific resource availability into hardware-independent dynamic service abstractions. This will enable a flexible and efficient use of network resources while meeting application/customer needs.

The node architecture consists of a *node data plane* and a *node control plane*. The former implements the physical data switching, whereas the latter exchanges in-band and/or out-of-band signaling information with the network control and management planes and configures the data plane accordingly. To implement router programmability, the first choice is to keep the control functions separated from the forwarding functions in the node control plane, as suggested by the ForCES framework.

In particular, Fig. 2.1 refers to the case of interest here, that is a router capable of optical switching at different granularities, i.e. at the packet (OPS), burst (OBS) and circuit (OCS) level. The data plane includes an optical switching matrix

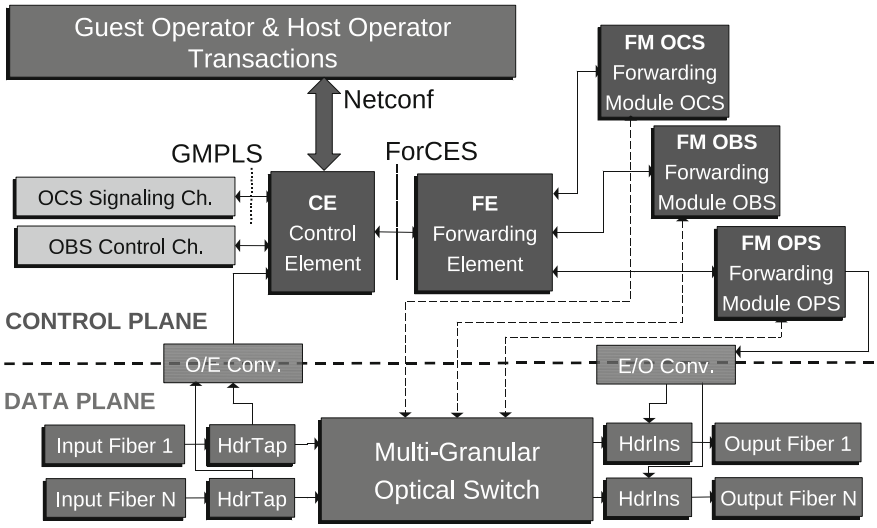


Fig. 2.1 Programmable router architecture, showing the node control plane building elements (CE, FE, FMs) and the interactions with node data plane, network control plane and management plane

connecting N input fibers to N output fibers, each carrying W wavelengths. The node control plane is further split into different elements, namely *Control Element* (CE), *Forwarding Element* (FE) and *Forwarding Modules* (FMs). The concepts of CE and FE are inherited from the ForCES architecture, whereas a further system modularization is introduced here by adding as many FMs as the different switching granularities available. This way, the FE can be kept hardware-independent.

The interactions with the network control plane and management plane are implemented by the entity located at the highest layer, i.e. the CE. More specifically, the management plane transactions—required by GOs to program the node with the desired functions—are performed by means of Netconf, due to its native remote router configuration capabilities. As for the network control plane, the CE processes information from in-band (OPS headers) or out-of-band (OBS and OCS control channels, the latter using GMPLS) signaling and performs the required admission and security checks before interacting with the FE to properly configure the node. The FE manages the logical forwarding operations according to the requests arriving from the CE and takes its decisions independently of the hardware resources available in the switch, thus providing actual separation between control and data planes. Then, it sends a request to the proper FM, based on the traffic needs, to configure the switching fabric. Each FM decides which requests coming from the FE can actually be satisfied according to the hardware availability and the state of physical resources. The FM drives the hardware devices to configure the switching matrix on the basis of the accepted requests and notifies success or failure to the FE.

This approach allows a virtualized description of the switch hardware resources, which can be used by the FE when performing the forwarding functions. In particular, a HO can build and install new FMs and allow GOs to activate them through the CE and FE interactions, whereas the FE will be in charge of creating the forwarding tables and managing the forwarding algorithms according to QoS and other needs identified by the CE.

2.4 Security Aspects in Programmable Routers

Any programmable network architecture raises several issues related to different security and reliability aspects, which can be addressed in different ways depending on the specific scenario considered [22–25]. This section describes such problems and proposes possible solutions with reference to two situations strictly related to the proposed approach, i.e. (i) when a GO asks the HO to activate a given programmable function and (ii) when a customer of that GO wants to actually use that function. Issues related to accounting mechanisms between HO and GO are not considered here.

Authentication. Whenever an external ISP wants to configure itself as a GO for a given service on top of the HO network, the HO starts a negotiation phase to authenticate the GO before allowing it to use the infrastructure. On the other hand, the GO must be sure to talk to the exact HO that it wants to contact. This reciprocal authentication problem can be tackled by adopting a typical Public Key Infrastructure (PKI) solution [26], where the two entities exchange their trusted signed certificates to identify themselves, but only after a *Guest Operator Service Level Agreement* (GO-SLA) has been established between them. Once a given GO has been allowed, its customers start to request the activation of the desired network services offered by their GO on top of the HO facilities. In this case, the HO must validate the authenticity of the request before allocating resources to the customer traffic. The solution this time depends on the kind of service requested and the related signaling protocol used. In the specific case of multi-granular switching services analyzed here, assuming a GMPLS-based control plane, the standard authentication mechanism provided by RSVP-TE could be adopted. According to this scheme, each signaling message exchanged between RSVP-TE-aware nodes may carry an *Integrity Object*, which includes an incremental sequence number and a message digest obtained by applying a keyed hash function, e.g. SHA-1, to the entire RSVP-TE message [27]. Such an authentication scheme is applied on a hop-by-hop basis, i.e. each intermediate RSVP-TE node must share a secret key with its neighbors and authenticate the messages exchanged with them. Therefore, when a GO customer requests a given service, e.g. to establish an optical circuit between two end-points, the authentication can be performed at the first node that receives the PATH message used to establish a new circuit. The same authentication method is kept also inside the network to avoid forged signaling messages to activate unauthorized services or to maliciously close existing ones.

Authorization. When an authenticated GO wishes to activate a given programmable function, it must be one of those allowed by the current GO-SLA. At the same time, when a customer of a GO requests a given service, it must be one of those available to the customer profile and must not exceed the maximum number of such services already activated. This problem can be solved by applying a white list approach at both levels. The HO keeps a white list of the GOs and, for each of them, the set of authorized programmable functions. In case a GO tries to activate a network function not explicitly allowed by the GO-SLA, the HO does not find it in the white list and then it rejects the request. A similar approach is used in the customer service request. The HO keeps, for each GO, a white list with the type and number of services allowed to the specific customer profile. It also maintains a list of the addresses of the ingress routers of the authorized GO. In case a customer asks for a service either not included in the corresponding profile or whose maximum allowed number of instances has been reached, the HO rejects the request.

Integrity. The information exchanged by the HO with the GOs or with their customers must always be checked for possible alterations, which could happen either accidentally or intentionally. Therefore, integrity is another crucial aspect of the proposed model. During the reciprocal authentication phase between HO and GO, the two entities must agree on the integrity check mechanism that they want to enforce in the following information exchange. A typical approach is to digitally sign each transaction using the same PKI solution adopted in the authentication phase. As for the signaling messages exchanged between the HO and the authenticated customers, the underlying protocol should be in charge of the integrity check. In the use case analyzed here, integrity check is performed by RSVP-TE along with the message authentication phase, since the hash function is applied to the whole RSVP-TE message. An alteration of any part of the message will cause the message digest computed by the receiver to be different from the one included in the Integrity Object by the sender and then the message to be discarded. The message digest computed on the altered message cannot replace the original one since the key is known only to sender and receiver.

Confidentiality. In the proposed architecture, the amount of information exchanged that must be kept confidential depends on the purpose of the information itself. Any transaction that involves the transmission of critical information, like for instance the distribution of the keys used for RSVP-TE authentication, must be encrypted using a robust method. PKI solutions are typically used for this purpose. Any other confidentiality requirement to be applied to the data-plane must be enforced either by a specific network service (e.g. by activating a secure tunnel based on IPsec) or by end-user applications (e.g. connections based on SSL/TLS).

Protection and availability. The services offered by the HO should be available 24-7, as long as they are included in the GO-SLA. Therefore, the HO should enforce suitable protection mechanisms such that any service interruption caused by accidental events (e.g. equipment malfunctioning, cable cuts, natural disasters) should be minimized. This means that some form of redundancy and backup resource allocation must be deployed, like the protection and restoration techniques defined

by GMPLS for the case analyzed here. Another important cause of loss of availability is represented by malicious denial-of-service attacks, where a huge amount of fake requests are sent in a short time in order to make the system crash and block any other legitimate request. Typical intrusion detection/prevention solutions may be enforced to deal with this problem.

2.5 Enforcing Secure Multi-Service Provisioning

This section describes how to design the node control plane to provide multi-service programmability while dealing with some of the security aspects mentioned above. In particular, the focus is on the CE of a HO router offering optical packet and circuit switching capabilities to the GO customers. This sample case-study could be extended to a general set of programmable services. The network control plane is assumed to be based on GMPLS, with RSVP-TE as the signaling protocol to set-up and tear-down optical circuits [28]. RSVP-TE messages of interest here are:

- *PATH*. Circuit set-up request, which travels along the path from the sender end-point to the receiver end-point. A *Label Set Object* is included where each node provides the next hop with the wavelengths available to set-up the circuit.
- *RESV*. Circuit set-up confirmation, originated from the receiver and forwarded by each intermediate node to the previous hop toward the sender to inform it that the hardware resources needed to set-up the circuit have been properly configured. A *Label Object* is included where each node provides the previous hop with the wavelength to be used by the circuit.
- *PATHERR*. Error message sent from a node to the upstream hops to notify that the circuit requested with a PATH message could not be established.
- *RESVERR*. Error message sent from a node to the downstream hops to notify that the circuit confirmed with a RESV message could not be established.
- *PATHTEAR*. Circuit tear-down message sent by the sender to release the resources along the path to the receiver.

All RSVP-TE messages include a *Session Object* to specify the optical circuit they refer to. They also include an Integrity Object to enforce secure transactions, as explained in Sect. 2.4. Figure 2.2a describes the internal organization of the CE. A message coming from the OCS signaling channel is received by the RSVP-Manager, which extracts information inside and pass it to the CE Security Module (CE-SM). The CE-SM checks whether the IP addresses of sender, receiver and previous hop belong to its customer whitelist, filled by the authenticated GOs. If not, the node discards the message due to unauthorized access. Otherwise, the CE-SM checks the message integrity using the Integrity Object, which carries a key identifier and a message digest. The CE-SM uses the sender IP address and the key identifier to uniquely determine the HMAC algorithm (SHA-1 is used here) and the key to compute the message digest. If the result matches the message

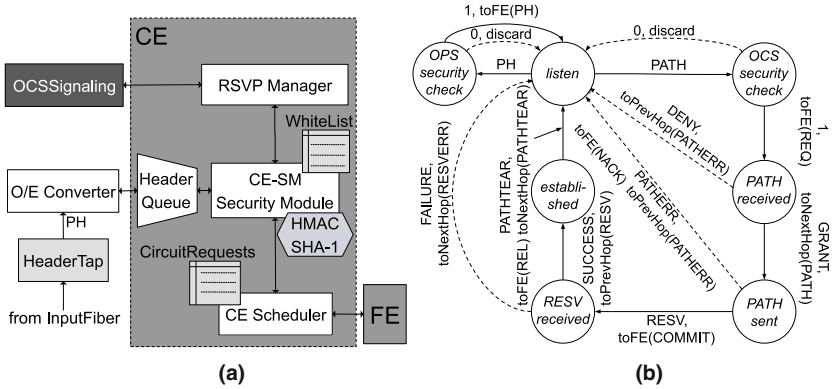


Fig. 2.2 **a** CE internal organization including security module; **b** Finite State Machine of a CE execution thread receiving either a packet header or a PATH message

digest field in the Integrity Object, the message is authentic, otherwise the CE-SM drops the message due to integrity error. Once all the security checks have been passed, the message is handed to the CEScheduler module, which updates a CircuitRequests table and then sends a request to the lower-level FE to perform the forwarding operations, such as resource availability check, reservation/release etc. Packet headers (PH) are subject to similar security checks to verify header integrity and, if required, sender authorization. Then they are passed to the FE via the CEScheduler.

The implementation of the CE operations described above is based, as far as a single execution thread, on the Finite State Machine (FSM) of Fig. 2.2b. The following set of communication messages between CE and FE is defined, which will be further extended to be compliant with the ForCES standard protocol:

- *REQ*, sent from CE to FE to check if it is possible to accept a new circuit, i.e. if the maximum number of allowed circuits has not been reached;
- *GRANT/DENY*, sent from FE to CE when the circuit can/cannot be accepted;
- *COMMIT*, sent from CE to FE to ask to reserve the resources (through FM OCS) to establish the new circuit;
- *NACK*, sent from CE to FE to inform that the circuit request is no longer valid;
- *SUCCESS/FAILURE*, sent from FE to CE to notify that the hardware resources have/have not been properly configured;
- *REL*, sent from CE to FE to release the resources dedicated to a circuit.

The state machine of a CE consists of the following states:

listen. The FSM is in the *listen* state until it receives either a Packet Header (PH) or a PATH message, after which it moves to the corresponding security check state.

OPS security check. The PH is subject to the required security checks from the CE-SM. If they succeed (1), the PH is sent to the FE to be scheduled, otherwise (0) the packet is discarded. In both cases, the FSM returns to the *listen* state.

OCS security check. If the message is not authorized by the CE-SM (0), the PATH is discarded and the FSM goes back to the *listen* state. Otherwise (1), a REQ message is sent to the FE and the FSM moves to the *PATH received* state.

PATH received. The FE checks whether or not it is possible to configure a new circuit. If not, the FE replies with a DENY message, the FSM turns back to the *listen* state and a PATHERR message is sent to the previous hop. Otherwise, the FE replies with a GRANT message and the CE forwards the PATH message to the next hop, while the FSM moves to the *PATH sent* state, waiting for a RESV message.

PATH sent. If a PATHERR is received, the FSM returns to the *listen* state and the CE forwards the PATHERR to the previous hop and a NACK to the FE. If a RESV is received, the CE sends a COMMIT message to the FE to configure the hardware devices (through the FM OCS) and the FSM moves to the *RESV received* state.

RESV received. If the FE replies with a FAILURE, meaning that the resources to set-up the circuit are no longer available, the FSM returns to the *listen* state and a RESVERR is sent to the next hop. If the FE replies with a SUCCESS, the FSM moves to the *established* state and the CE forwards the RESV to the previous hop.

Established. The node is ready to accept circuit traffic and waits. When the CE receives either a PATHTEAR or a RESVERR (not shown in Fig. 2.2b) from the previous hop, the FSM moves to the *listen* state and a REL message is sent to the FE. The PATHTEAR or RESVERR is then forwarded to the next hop. Since RSVP-TE works in soft-state mode, PATH and RESV messages must be periodically sent to keep the current state in the intermediate nodes. Figure 2.2b does not show the existing transitions from each state to the *listen* state due to time-out expiration.

2.6 Experimental Validation

A software router test-bed has been implemented using the *Click!* modular router framework [29] to emulate the described programmable router functions. The standard and enhanced *Click!* elements described in [30] are here extended for the multi-service support, by adding OCS capabilities. CE functionalities discussed above are implemented. The experimental set-up includes an OPS/OCS node equipped with $N = 2$ input/output fibers with $W = 4$ wavelengths and the required traffic generators. Time slotted operations are assumed for OPS traffic with slot duration equal to the time unit, while OCS signaling is asynchronous. Resources for up to two simultaneous circuits are available to OCS traffic. At time $t_0 = 0.00$, the node forwards only packet traffic, with load $\rho = 1$ per channel. New circuit set-up requests are generated at times $t_1 = 294.98, t_2 = 504.96, t_3 = 707.95, t_4 = 1010.93$ and $t_5 = 2010.85$. The related RSVP-TE messages exchanged between the node and its previous hops and the Netconf transactions with the

No. .	Time	Source	Source Port	Destination	Dest Port	Protocol	Info
2	294.981088	172.16.2.1		192.168.102.1		RSVP	PATH Message. SESSION: IPv4-LSI
3	504.961799	172.16.5.1		192.168.105.1		RSVP	PATH Message. SESSION: IPv4-LSI
4	507.970776	10.50.51.2		10.50.51.1		RSVP	RESV Message. SESSION: IPv4-LSI
5	530.903264	192.168.10.67	43195	192.168.10.69	830	TCP	43195 > netconf-ssh [PSH, ACK]
6	530.943129	192.168.10.69	830	192.168.10.67	43195	TCP	netconf-ssh > 43195 [ACK] Seq=1
7	531.307659	192.168.10.69	830	192.168.10.67	43195	TCP	netconf-ssh > 43195 [PSH, ACK]
8	531.307777	192.168.10.67	43195	192.168.10.69	830	TCP	43195 > netconf-ssh [ACK] Seq=1
9	707.954270	172.16.0.1		192.168.100.1		RSVP	PATH Message. SESSION: IPv4-LSI
10	710.959809	10.50.50.2		10.50.50.1		RSVP	RESV Message. SESSION: IPv4-LSI
11	1010.93317	172.16.7.1		192.168.107.1		RSVP	PATH Message. SESSION: IPv4-LSI
12	1010.93560	10.100.50.1		10.50.51.1		RSVP	PATH ERROR Message. SESSION: II
13	1410.89875	172.16.0.1		192.168.100.1		RSVP	PATH TEAR Message. SESSION: IPv
14	1710.87193	172.16.5.1		192.168.105.1		RSVP	PATH TEAR Message. SESSION: IPv
15	2010.84511	172.16.1.1		192.168.101.1		RSVP	PATH Message. SESSION: IPv4-LSI
16	2013.85445	10.50.50.2		10.50.50.1		RSVP	RESV Message. SESSION: IPv4-LSI

Fig. 2.3 Capture of the RSVP-TE and Netconf messages in the experimental node validation

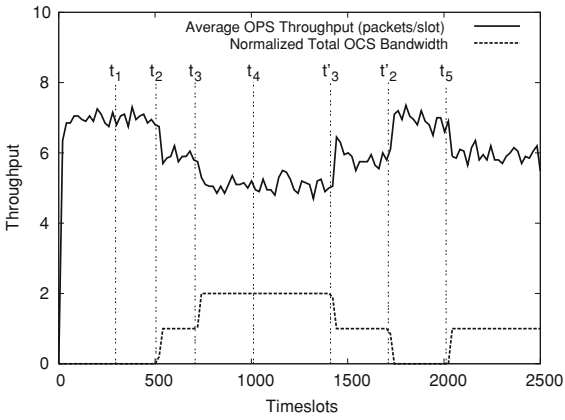


Fig. 2.4 Measured OPS/OCS throughput in the experimental node validation

GO are shown in Fig. 2.3. Circuit traffic is generated by constant bit rate sources transmitting at full rate. The average OPS throughput in terms of forwarded packets per time slot and the total normalized OCS bandwidth, with a sample period of 20 time units, are shown in Fig. 2.4.

When no circuits are active, the OPS throughput is affected only by the packet loss rate due to packet contentions. The first PATH message at time t_1 is ignored, being it generated by an unauthorized user. The second PATH message at t_2, after authorization check, is followed by the corresponding RESV which successfully completes circuit set-up. The OPS throughput decreases, while the OCS bandwidth reaches 1. A secure GO-HO transaction takes place at time 530.90, when a new customer is authorized. The new customer asks for a circuit at t_3. The OPS throughput further decreases while the OCS bandwidth doubles. The PATH message at t_4 from an authorized sender is rejected with a PATHERR message, since the maximum number of two admissible circuits is overcome. PATHTEAR messages at times $t'_3 = 1410.90$ and $t'_2 = 1710.87$ release the established circuits and cause the OPS throughput to raise again. At t_5 a new circuit request can be accepted.

2.7 Conclusions

Security aspects of a multi-service programmable router have been presented and tested in a software router framework. Multi-granular programmable service support with security features is obtained as circuit and packet switching by applying control and forwarding function separation within a modular router architecture. Control element implementation and its interfacing with the forwarding element to configure lower level forwarding modules has been proved through the correct set-up of circuits based on RSVP-TE signaling, on top of emulated optical switching matrix. The framework presented can be extended to take further enhanced services and security issues into account and to implement standard compliant protocols. The software test bed approach has been shown to be an effective development environment for router functionalities and concepts even in advanced technology contexts.

Acknowledgements This work was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe"), funded by the European Commission through the 7th ICT-Framework Programme.

References

1. European Commission—Directorate-General Information Society: Future Internet Assembly 2010—Conference Report, Valencia, Spain, April 2010, http://www.future-internet.eu
2. Wosinska, L., Simeonidou, D., Tzanakaki, A., Raffaelli, C., Politi, C.: Optical networks for the future internet: introduction. IEEE/OSA J. Opt. Commun. Netw. **1**(2), FI1–FI3 (2009)
3. Van der Merwe, J., Kalmanek, C.: Network programmability is the answer! Workshop on Programmable Routers for the Extensible Services of Tomorrow (PRESTO 2007), Princeton, NJ, May 2007
4. Chen, T.M., Jackson, A.W.: Active and programmable networks (guest editorial). IEEE Netw. **12**(3), 10–11 (1998)
5. Rooney, S., et al.: The Tempest: a framework for safe, resource assured, programmable networks. IEEE Commun. Mag. **36**(10), 42–53 (1998)
6. Huard, J.-F., Lazar, A.A.: A programmable transport architecture with QoS guarantees. IEEE Commun. Mag. **36**(10), 54–62 (1998)
7. Tennenhouse, D.L., et al.: A survey of active network research. IEEE Commun. Mag. **35**(1), 80–86 (1997)
8. Calvert, K.L., Bhattacharjee, S., Zegura, E., Sterbenz, J.: Directions in active networks. IEEE Commun. Mag. **36**(10), 72–78 (1998)
9. Alexander, D.S., et al.: The SwitchWare active network architecture. IEEE Netw. **12**(3), 29–36 (1998)
10. Wetherall, D., Legedza, D., Guttag, J.: Introducing new Internet services: why and how. IEEE Netw. **12**(3), 12–19 (1998)
11. Workshops on Programmable Routers for the Extensible Services of Tomorrow, PRESTO 2007, Princeton, NJ, May 2007; PRESTO 2008, Seattle, WA, August 2008; PRESTO 2009, Barcelona, Spain, August 2009
12. Yanget, L., et al.: Forwarding and control element separation (ForCES) framework. IETF RFC 3746, April (2004)

13. Doria, A., et al.: Forwarding and control element separation (ForCES) protocol specification. IETF RFC 5810, March (2010)
14. Wang, W., Dong, L., Zhuge, B.: Analysis and implementation of an open programmable router based on forwarding and control element separation. J. Comput. Sci. Technol. **23**(5), 769–779 (2008)
15. Haleplidis, E., et al.: A Web Service- and ForCES-Based Programmable Router Architecture. Lecture Notes in Computer Science, vol. 4388, pp. 108–120. Springer, New York (2009)
16. Enns, R.: NETCONF configuration protocol, IETF RFC 4741, December (2006)
17. Zervas, G.S., et al.: Multi-granular optical cross-connect: design, analysis and demonstration. IEEE/OSA J. Opt. Commun. Netw. **1**(1), 69–84 (2009)
18. Zervas, G., et al.: Programmable multi-granular optical networks: requirements and architecture. In: Proceedings of Broadnets 2009 Madrid, Spain, September (2009)
19. Callegati, F., et al.: SIP-empowered optical networks for future IT services and applications. IEEE Commun. Mag. **47**(5), 48–54 (2009)
20. Martini, B., Martini, V., Baroncelli, F., Torkman, K., Castoldi, P.: Application-driven control of network resources in multiservice optical networks. IEEE/OSA J. Opt. Commun. Netw. **1**(2), A270–A283 (2009)
21. Qin, Y., et al.: Service-oriented multi-granular optical network testbed. In: Proceedings of OFC 2009, San Diego, CA, March (2009)
22. Alexander, S., et al.: Safety and security of programmable network infrastructures. IEEE Commun. Mag. **36**(10), 84–92 (1998)
23. Gao, J., Steenkiste, P.: An access control architecture for programmable routers. In: Proceedings of IEEE OPENARCH 2001, Anchorage, AK, April (2001)
24. Murphy, S., Lewis, E., Puga, R., Watson, R., Yee, R.: Strong security for active networks. In: Proceedings of IEEE OPENARCH 2001, Anchorage, AK, April (2001)
25. Alarco, B., Sedano, M., Calderon, M.: Multidomain network based on programmable networks: security architecture. ETRI J. **27**(6), 651–665 (2005)
26. ITU-T Recommendation X.509: Information Technology—Open Systems Interconnection—The Directory: Public-Key and Attribute Certificate Frameworks, August (2005)
27. Baker, F., et al.: RSVP cryptographic authentication. IETF RFC 2747 January (2000)
28. Berger, L. (ed.): Generalized multi-protocol label switching signaling resource reservation protocol-traffic engineering (RSVP-TE) extensions. IETF RFC 3473, January (2003)
29. Kohler, L., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.F.: The click modular router. ACM Trans. Comput. Syst. **18**(3), 263–297 (2000)
30. Cerroni, W., Raffaelli, C., Savi, M.: Software emulation of programmable optical routers. In: Proceedings of HPSR 2010, Dallas, TX, June (2010)

Chapter 3
Secure Pseudonymous Access to Overlay Data Dissemination Network

Anna Del Grosso, Marco Listanti, Andrea Baiocchi and Matteo D'Ambrosio

Abstract New paradigms for the Internet architecture evolution towards a data dissemination oriented have been largely proposed. Scott Shenker et al. (SIGCOMM 2007) define the basic principles desirable for a dissemination network, but they do not suggest how to realize them. We introduce the Secure Pseudonymous Access (SPA), an access protocol suitable for every dissemination network. The main goal is to couple QoS constraints in terms of data reliability and secure access with easiness of service use, by removing most of administrative and user initialization burdens. The key issue is the user traceability, i.e. the possibility to tie together (in a provable way) the actions performed by the same user entity, along with pieces of data uploaded into the network to be shared with others. The signalling procedures of SPA are defined and the security issues are discussed; finally we describe a test bed implementation of SPA and give an estimate of procedure complexity.

Keywords Security · Overlay network · Pseudonyms · QoS

A. Del Grosso (✉) · M. Listanti · A. Baiocchi
INFOCOM Department, University of Rome "Sapienza" Via Eudossiana, 18-00184 Rome, Italy
e-mail: delgrosso@infocom.uniroma1.it

M. Listanti
e-mail: listanti@infocom.uniroma1.it

A. Baiocchi
e-mail: baiocchi@infocom.uniroma1.it

M. D'Ambrosio
e-mail: matteo.dambrosio@telecomitalia.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_3,

3.1 Introduction

When Internet was first developed, it was intended to be a data-sharing network for the military and research facilities. No one foresaw its growth as a communications powerhouse. In the last years, in conjunction with the increasing number of users, the scope of Internet has changed and its usage mainly concerns the download of specific contents, instead of the communication between specific hosts in the network; this is reflected in a new communication paradigm focused on data (data-oriented), instead on host-to-host communications (host-oriented). In the host-oriented view the network has the only role to carry on the packets to the destination; on the contrary, in the data-oriented paradigm, the main focus is on data retrieval. Users do not care of the data location, but they need the data security in terms of integrity, consistency and completeness. These thoughts bring to a new vision of Internet as a "*Dissemination Network*", as defined by Van Jacobson in [1], where the data matters but not who gives it to you. In this paradigm, it is crucial to take care of the different security aspects: for example the data owner has to know which users download its piece of data, while the user itself has to be sure the received data is the requested one. Specifically, we need to guarantee the data *Persistence, Availability and Authenticity* [1]. In the seminal paper [1] some guidelines to realize an efficient dissemination network are defined: (1) data must be requested by *name*, using any and all means available, (2) any entity receiving the request and having a valid copy of the data can respond to the request, (3) the returned data must be signed, and optionally secured, so that hte data integrity and the association with the name can be validated. According to these guidelines, the concept of Secure Pseudonymous Access (SPA) can be defined as follows: an access protocol for every dissemination network, able to guarantee the anonymity but also the accountability of users inserting the data in the dissemination network. Namely, each labelled content is associated to some identity who cannot repudiate its content nor its publication. This way the polluters or the unreliable information providers can be detected and isolated. Moreover the above access protocol seems to be very flexible since it does not need any administrative procedure based on contracts and/or formal statements of legal user identity. It aims at protecting the data integrity and authenticity, yet preserving the anonymity of users. For these reasons, SPA defines the users registration using *pseudonyms* valid only in the context of the network in which they operate; the content in the DHT is associated to the user pseudonym that registers it. In our view, the dissemination network is the Internet evolution, that is to say a common infrastructure in which different actors operate to provide different services and, specifically, different contents.

Related to this access procedure, the major contribution of this work is the definition of signalling procedure to register user's pseudonym in the dissemination network and to upload/retrieve data (Register and Find procedures). These procedures are very incisive because they do not depend neither on the content the user wants to download nor on the actor who wants to publish it. In particular we introduce the use of the *tickets* to authorize a specific action (register or find).

The tickets are issued by the correspondent actor that could be a Content Provider or a registered user having access to the infrastructure. Register and Find procedures are stated in an enough general context to be suitable for *any* dissemination network architecture, that is these procedures are *independent* on the protocols used in the overlay network. The signalling procedure has been implemented and tested in a lab test bed and a preliminary evaluation of processing complexity was carried out.

Various architectures have been recently proposed to overcome current Internet flaws and to support the concept of dissemination network: Internet Indirection Infrastructure [2] (i3), Host Identity Protocol [3] (HIP), Delegation Oriented Architecture [4] (DOA), Split Name Forwarding [5] (SNF). However all these solutions present several drawbacks, such as the lack of security and the complete modification of Internet procedures. The architecture that best reflects the requirement of a dissemination network appears to be DONA [6]. Data Oriented Network Architecture (DONA) is a network architecture which operates over the IP layer with the scope of the data dissemination in Internet. To achieve the Van Jacobson's principles [1], DONA defines *Principals*, logical entities able to publish the contents into the dissemination networks, and the association between Principals and the content to be published. DONA uses the *route−by−name* paradigm for names resolution, so that a request can be routed toward the nearest copy of data; the underlying point-to-point IP layer is unchanged and two new primitives are defined over it: Register(D) and Find(D), respectively to register or request a piece of data D.

Paper organization is as follows. In Sects. 3.2 and 3.3, we detail the SPA paradigm in the context of dissemination network, giving a formal definition and describing briefly its procedures, while in Sect. 3.4 we describe the test bed implemented in our laboratory closing up with the time estimation of SPA's procedures.

3.2 SPA Concept and Motivation

DONA defines the basic principles desirable for a dissemination network, but it does *not* suggest how to realize them. We apply the concept of the SPA to propose a concrete realization of DONA's principles. Our proposal is secure and flexible at the same time: it aims at eliminating some rigid procedures based on formal statements (such as the identification through certificates issued by recognized certification authorities) in favour of some flexible procedures based on tickets issued by particular *registration authorities*. The use of the pseudonymous access allow us to represent the user in a way specific to the dissemination network (through user registration and profiling) without relying on any administrative procedure based on contracts or formal statement regarding the legal identity of the user. If we want to enrich the dissemination network with QoS constraints and, at the same time, leave the door open to the paradigm of the pseudonymous access, we need some form of *accountability*, that is to say we need to trace the users'

activity and tie together their actions. One of the main requirement is an easy-to-go access to information retrieval and contribution, motivated by the demonstrated booster effect of a mass sharing of information, which widens the offered spectrum of contents and enables the peers information sharing, thus making room to a large variety of different business and social initiatives. The down side of the liberality in the access is the potentially low quality and unreliability of the shared information. Architectures like DONA try to define mechanisms to assess "authenticity" and integrity of the shared information, but they leave little defence against the malicious users that deliberately introduce corrupted or false information in the network. To address this issue, we define an initial registration procedure, in which we initialize the security parameters and provide some form of authentication. Since non repudiation is a key security property in this context, we resort to public-key cryptography, so that we can rely, for istance, on the techniques of [7]. The accountability guaranted by SPA provides the possibility to insert a reputation system in the dissemination network to evaluate the behaviour of the various principals.

In our proposal, the SPA scenario is composed of some logical functional entities. We refer to the general architecture of a dissemination network as an *Information Centric Network* (ICN); essentially it is an overlay network for content routing. The *Access nodes*, denoted *Authoritative Resolvers* (ARs), take care of the data Register and Find requests and check for the authorization requirements when delivering the requested data. Authoritative Resolvers functionality could be managed by different units who are in charge of providing the accountability feature, besides the basic data security features. The *ICN client* is the user requesting service to ICN. We define two types of users: the Principal and the Consumer. The *Principal* P is the producer of a piece of data D, which is made available to other users through one or more ICN management units; according to the SPA procedures, P is responsible for it and can not repudiate it. The *Consumer* is the generic user accessing the network looking for a particular content; this content is always associated to a specific Principal who guarantees the registered data. The *Registration Server* (RS) is the logical entity used to register an ICN client and issue the tickets to insert or download a specific content. It is important to underline that we will make use of a different RS for each management unit, that is to say for each service provider who wants to grant access to the ICN. Finally, there is a *Data Information Data Base* (DIDB) to store users and data's credentials.

Let $H(\cdot)$ be a collision resistant hash function, let (pk, sk) be a public/private key pair and let Π = (sign, Vrfy) be a digital signature scheme universally unforgeable under adaptive chosen message attacks (UF-CMA). Following the $hash - and - sign$ paradigm[1] [8] we define the following signing function on a string $D \in \{0,1\}^*$:

[1] It is well known that if $H(\cdot)$ is collision resistant and Π is UF-CMA then the signing function defined in (3.1) is UF-CMA in the random oracle model.

$$\sigma(D) = \mathsf{sign}_{sk}(H(D)). \tag{3.1}$$

Given a pair $(D, \sigma(D))$, the signature is accepted as valid if and only if $\mathsf{Vrfy}_{pk}(H(D), \sigma(D)) = 1$. Each Principal is associated with (pk_P, sk_P), a self-generated public/private key pair, and computes:

$$P = \mathsf{sign}_{sk_P}(pk_P). \tag{3.2}$$

The data registered into an ICN domain are then structured as a triple $\langle \mathsf{label}, D, \sigma \rangle$. where label is a label that uniquely identifies that piece of data inside the entire ICN, D is the data itself, and σ is an integrity authenticator of the data (e.g. signature of D made by the owner of D, i.e. the Principal P that uploaded the data into the ICN). To associate data and its owner P, the field label in turn is structured as $\langle P, L \rangle$. P is an ICN-wide unique value representing the Principal's identity, as defined in (3.2). L is the data label, containing a footprint of D (e.g. the hash value $H(D)$) and an authorization field A, with a description of *requirements* needed to download D (e.g. being part of a given closed user group, having paid a certain amount, having a given coupon, none in case of a fully public and open document).

A user requesting a piece of data to a given domain, sends the request to any of the access nodes (Find procedure). The data is fetched to the AR, the field A is read and the user credentials are checked by running a specific protocol between the AR and the user. We have separated and highlighted the RS as a new functional unit (which can even be physically separate and remote with respect to the AR). Centralizing the RS function at domain level can simplify the AR architecture. Security related informations for each piece of data can either stay with the data itself and checked (whenever the piece of data itself is requested), or they can be placed in the DIDB of the RS and dealt with as a cache, to speed up the overall fetch of the data with security checks.

To sum up, RS accepts the requests from Principals and users to register them in its domain. Principals ask authorization to register a piece of data and the RS decides to grant or not this authorization according to the presented credentials. Besides RS agrees with Principals on users' requirements to request its specific object and updates the DIDB accordingly. When a user wants to download a specific piece of data, the RS, being able to access the DIDB, carries out the procedure to grant the authorization according to the specific requirements.

3.3 SPA Procedures

In this section we describe the Register and Find procedures, according to the proposed SPA. In what follows, RS is the *Registration Server*, P is the *Principal*, C is the *Consumer* and AR is the *Authoritative Resolver*. Formally, we can define a Principal as a pair $\langle P, \mathsf{Keys} \rangle$, where P is a unique user identifier as defined in 2 and

Keys $= (pk_P, sk_P)$ is the public/private key pair associated with that identity. A new user self-generates his own private/public key pair and shares the value of P and the public-key with the ICN operator. The private key must never be disclosed to any entity other than the user who generated it, since it is used to sign the data that P wants to publish. Procedures are based on tickets that Registration Server grants to the different users; field `Type` can contain value R indicating it is a "registration" ticket, or value F indicating it is a "retrieval" ticket; the field `Scope` clarifies the object that is under the action of the `Type` field and can contain the value E indicating it is an "entity", or the value C indicating it is a content.

3.3.1 Register Procedure

Register is the procedure by which a user, after becoming Principal, can register a piece of data in the ICN. According to the SPA, this procedure has to guarantee the accountability of Principals and the property of non repudiation of the actions on the registered content. It is composed of three logical steps; the first one has to be executed only one time, while the second and the third steps have to be executed each time a new content is going to be published: (1) *Becoming Principal (initialization step)*—a client who wants to publish a piece of data in the network has to register its identity in one of the ICN management domains and, thus become Principal (P); (2) *Being authorized to publish data*—P has to be authorized to publish a content; this authorization is given after the analysis of its credentials in relation to the preliminary domain's negotiation; (3) *Inserting data in the ICN*—P can insert a piece of data in the network, after presenting the received authorization to the AR.

Whereas the goal of the first two steps is user registration as a Principal and authorization to publish a piece of data, it is in the third step that content is actually registered in the ICN. In each step, it is fundamental to guarantee integrity and authenticity of the exchanged messages and to ensure that the communication flow involves always the same entities (*flow control*). To obtain the former, all the messages are signed using the private key of the relative entity, while to obtain the latter, we insert random numbers in the messages (replicated in session messages) as to guarantee flow control [9, 10].

Describing the procedure in details, a user registers its identity as P to the RS and receives from it an *Existing Ticket*, signed by RS's private key, that *certifies its public-key*: when the Principal wants to access the ICN, it should present this ticket to be authenticated. Specifically, every entity, trusting in RS and having RS's public key, could verify P's identity through this ticket. As soon as a client becomes Principal, he/she can publish data over the network. Before a content can be registered in the ICN, P must request the specific authorization. This authorization is issued by the RS after checking if P has the right credentials. The authorization consists in a *Data Registering Ticket* that P has to deliver to the

AR, the access node toward the ICN's routing architecture. At the end of this step, P has obtained the authorization to publish a piece of data and can contact the AR node to complete the registration procedure. AR has not any information about the registered Principal, but it can decide to authorize or not the registration, on the basis of the received ticket and the trust it has in the issuer entity. If all checks have been positive and P has been authenticated, AR proceeds to register content $\langle P, L \rangle$ in the ICN routing layer.

3.3.2 Find Procedure

Find is the procedure by which a client, after becoming a registered Consumer C, can request data from the ICN. Recall that there is not any kind of restriction to become an user and the registration is used to provide accountability.

Similarly to the Register procedure, the Find procedure is composed of three logical steps; the first one has to be executed only one time, while the second and the third steps have to be executed each time a content is going to be requested: (1) *Becoming a Consumer (initialization step)*—the user registers its identity in a management domain; (2) *Being authorized to request data*—after checking that the user holds the appropriate credentials, he is authorized to request the data; (3) *Requesting ICN data*—having the authorization, the user can now request the content.

Also in this procedure we need to be sure that the integrity and the authenticity of the exchanged messages is guaranteed and to ensure some form of *flow control*. To do this, during the communication, two secret-keys are shared: one key will be used to generate HMAC tags of the exchanged messages (to get some form of mutual authentication), the other key will be used to cipher the confidential informations using a symmetric-key encryption algorithm.

Notice that to register his identity in the future RS, C has not any secret shared with the ICN domain. Thus the Consumer have to choose a username and a password π to be identified by a pair (username, π) in the ICN domain. Also notice that, after becoming a Consumer and before requesting the authorization to find a content, C must know the label associated with the desired content. Since it is very important to obtain a valid label to guarantee content authenticity and integrity, this issue is crucial. Here we do not deal with this task and we suppose that C has received a valid label (say $\langle P, L \rangle$) from an authentic directory server and can contact RS to request the authorization for the content $\langle P, L \rangle$. This authorization is given to the Consumer in the form of a *Data Request Ticket* and allows him to contact the AR and request the content $\langle P, L \rangle$. In the last step of the *Find* procedure, C must forward its request to the AR. After validating the request and the received ticket, the AR sends a "look-up" query into the ICN routing network and waits for the requested data. Finally, C can download the content and verify its authenticity and integrity. Specifically C receives, through the routing layer, the

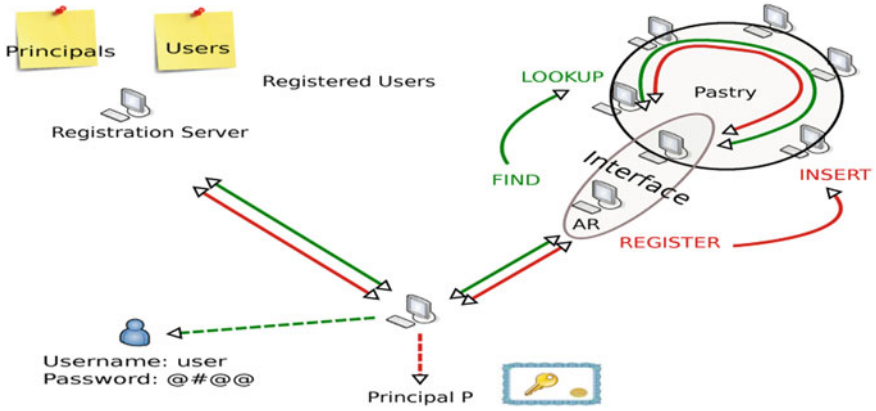


Fig. 3.1 Global picture of our test bed. Pastry has been implemented using [12] software

triple (D, pk_P, σ) made of the data content, P's public-key and P's signature on D. Using the the proper pair $\langle P, L \rangle$, the user can validate pk_P and then verify σ.

3.4 Test Bed Description

We have implemented all the logical functionalities defined in SPA in a test bed (see Fig. 3.1 for a global picture). To set-up the test we implemented the four logical nodes in SPA (namely the RSs, the ARs, Principals and Consumers) and detailed all the step of Register and Find The routing overlay of the ICN network is based on Pastry [11], a self-organizing overlay network used to support a variety of peer-to-peer applications, including global data storage, data sharing, group communication and naming. Specifically, we have used FreePastry, an open-source implementation of Pastry developed by Rice University [12]. FreePastry specifies two function, insert$(\cdot)$ and lookup$(\cdot)$, respectively, to register a new content in Pastry DHT and to retrieve a registered content. All the scheduled security mechanisms have been implemented, from ciphering or signing to HMAC's processing; in particular a ticket mechanism has been developed, to manage ticket creation, signing and transfer.

After developing all the signalling procedures of SPA, we took care of the integration between these procedures and FreePastry defining an *interface*: this is a sort of gateway used to "translate" the last step of Register and Find primitives into insert$(\cdot)$ and lookup$(\cdot)$ commands to be used by Pastry.

The SPA procedures have been implemented choosing specific values for the fields in the exchanged messages. Then we used out test bed to evaluate the complexity of the operations defined in SPA, with particular attention to the *Registration Server* and the *Authoritative Resolver*. By complexity, we mean

Table 3.1 Time estimates for the main cryptographic operation

Operation	Value	Estimate
RSA encryption (1024-bit key)	T_{RSA}	0.04 ms
RSA signature generation (1024 bits)	T_{σ}	0.68 ms
RSA signature verification (1024 bits)	T_{v}	0.04 ms
AES/CBC (128-bit key)	T_{AES}	$137 \cdot 2^{23}$ bit/s
SHA-1 (1024 bit key)	$T_{\mathsf{SHA-1}}$	$216 \cdot 2^{23}$ bit/s
HMAC (using SHA-1)	T_{HMAC}	$217 \cdot 2^{23}$ bit/s

the time a single server need to process Register and Find queries. To detail the analysis, we give the size of the basic values used:

- Any signature has length 1024 bits. This value has been chosen, considering the size of RSA signature, the most used digital signature scheme.
- The hash function $H(\cdot)$ is implemented using SHA-1. Thus the fingerprint of each message is 160 bits long.
- HMAC tags have the same size of the underlying hash function (160 bits using SHA-1).
- Nonce values have length 160 bits, and are produced using a cryptographic Pseudo-Random Number Generator.
- A digital certificate is, usually, 8 Kbit long.
- A character is represented by 8 bits.

According to these assumptions, we can define the size of the main fields in the exchanged message:

- $\langle P, L \rangle$: According to (2) the value P is a signature and thus has length 1024 bits; the value L is obtained hashing data string and, so, it is 160 bit long.
- (username,π): Character fields are composed of 20 characters yielding 160 bits.
- Encrypted values: Using RSA, encrypted values are 1024 bits long.
- Authenticators: signatures are 1024 bits long, HMAC tags 160 bits.
- RSA keys: They are 1024 bits long.

The estimation of tickets size depends on their definitionticket definition and fields size; the `Type` and `Scope` fields are 2 bits long while the `LifeTime` field is 160 bits long (since it contains the starting and expiration dates): we get (1) 2212 bits for *Existing Tickets*, (2) 3396 bits for *Data Registering Tickets*, and (3) 2532 bits fot *Data Request Ticket*. Recall that any ticket has associated a signature (1024 bit).

To sum up, we have used RSA protocol (1024 bits keys) for asymmetric encryption/decryption and for signature generation/verification. SHA-1 is the hashing algorithm used also to produce HMAC tags. Finally we rely on AES (with CBC operational mode and 128 bits key) for symmetric encryption/decryption. To process security's operations, we have used speed benchmarks for some of the most commonly used cryptographic algorithms, described in [13] and summarized in Table 3.1. All were coded in C++, compiled with GCC 4.1.1 using

Table 3.2 Time estimates for the Register and Find procedures (ms)

Register procedure	P	RS/AR	Total
Becoming principal	0.847	2.123	2.97
Being authorized to publish data	0.889	2.764	3.653
Inserting data in the ICN	0.846	2.123	2.969
Find procedure	**U**	**RS/AR**	**Total**
Becoming a registered user	0.129	1.364	1.493
Being authorized to request data	0.172	2.047	2.219
Requesting ICN data	0.130	1.407	1.537

-O2 optimization, and ran on an AMD Opteron 2.4 GHz processor under Linux 2.6.18 in 64-bit mode.

Using the values in Table 3.1, assuming a data string D of length ℓ bits, we get the following estimates for the processing time of the main cryptographic operations:

- The time needed to evaluate $H(D)$ is $\frac{\ell}{T_{\text{SHA}-1}}$.
- The time needed for signature generation is $\frac{\ell}{T_{\text{SHA}-1}} + T_\sigma$.
- The time needed for signature verification is $\frac{\ell}{T_{\text{SHA}-1}} + T_v$.
- The time needed for AES encryption using CBC is $0.003\,\text{ms} + \frac{\ell}{T_{\text{AES}}}$.
- The time needed for HMAC tag generation is $0.0005\,\text{ms} + \frac{\ell}{T_{\text{HMAC}}}$

According to the procedures described in Sect. 3.3 and using the derived patterns from Table 3.1, a preliminary performance evaluation is given in Table 3.2.

Table 3.2 contains an estimate of the time (ms) spent by the users (either P or C) and servers (either RS or AR) to process the operation involved in the Register and Find procedures. The table gives the time needed by each part to perform each step in the procedures and also the total time required. Recall that the first two steps involve only the RS, whereas the third step involve the AR. Notice that the RS/AR has a non negligible processing time, much higher than the processing time of P/C. This difference is due to the different operations each entity must do. Specifically the RS has to compile the authorizations and sign all the exchanged messages; on the other hand P has to verify authorizations and (eventually) sign the messages. Finally C has the less processing time, because he does not rely on any asymmetric technique: C has to verify authorizations, encrypt/decrypt using AES and evaluate HMAC tags. The time needed for these operations is less onerous.

The results shown in Table 3.2 can be exploited to assess the processing load that servers can carry. For example, looking at the second column of Table 3.2, the time the AR is busy with a data upload/refresh procedure is 2.123 ms. Hence the maximum rate of this procedure, if an entire CPU is devoted to this specific task, is about 470 refresh procedures/s. If the average refresh time is 1 h, the upper bound of the manageable pieces of data is less then 1.7 million.

Acknowledgements The topic of this paper includes description of results of a research project carried out by INFOCOM Dept, "Sapienza" University of Rome on behalf of (and funded by) Telecom Italia S.p.A., who reserve all proprietary rights in any process, procedure, algorithm, article of manufacture, or other results of the project herein described. The first author is thankful to Daniele Venturi for helpful discussions on the write-up of this paper.

References

1. Jacobson, V.: If a Clean Slate is the solution what was the problem? Stanford "Clean Slate" Seminar (2006).
2. Stoica, I., Adkins, D., Zhuang, S., Shenker, S., Surana, S.: Internet indirection infrastructure. In: SIGCOMM '02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (2002)
3. Moskowitz, R., Nikander, P.: Host identity protocol architecture. Internet-Draft, IETF, Apr 2006
4. Walfish, M., Stribling, J., Krohn, M., Balakrishnan, H., Morris, R., Shenker S.: Middleboxes no longer considered harmful. In: OSDI'04: Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation, pp. 15–15. USENIX Association (2004)
5. Jonsonn, A., Folk, M., Ahlgren, B.: The split naming/forwarding network architecture. In: SNCNW '06: Proceedings of Swedish National Computer Networking Workshop (SNCNW), 2006
6. Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. SIGCOMM Comput. Commun. Rev. (2007)
7. Chaum, D., Antwerpen, H.V.: Undeniable signatures. In: CRYPTO89, pp. 212–217. Springer-Verlag (1990)
8. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman and Hall/CRC Press (2008)
9. Evans, D., Beresford, A.R., Burbridge, T., Soppera, A.: Context-derived pseudonyms for protection of privacy in transport middleware and applications. In: PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (2007)
10. Dang, X., Zhang, Y.: Hierarchical pseudonym-based signature scheme and self-generated pseudonym system in Ad Hoc networks. In: ICWMC '08: Proceedings of the 2008 the Fourth International Conference on Wireless and Mobile Communications,IEEE Computer Society, Washington, (2008).
11. Rowstron, A.I.T., Druschel, P.: Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg. Springer-Verlag (2001)
12. USA Rice University, Houston. Pastry implementation, http://www.freepastry.org. Accessed May 2008.
13. Cryptographic benchmark. http://www.cryptopp.com/benchmarks.html (2008). Accessed May 2008

Chapter 4
An Overlay Infrastructural Approach for a Web-Wide Trustworthy Identity and Profile Management

Maria Chiara Pettenati, Lucia Ciofi, David Parlanti, Franco Pirri and Dino Giuli

Abstract Scalability and trust-enabling capabilities in Web-wide Profile Management are two challenges addressed in this paper. An infrastructural theoretical framework is proposed and discussed, based on the InterDataNet (IDN) architecture aimed at supporting seamless data interoperability to facilitate sharing and exploitation of distributed and heterogeneous user profile information. This paper presents the grounding issues of a more detailed analysis in progress on this problem, to highlight the prospected advantages and the technical viability of the proposed approach. It is possible to apply the framework in cross-domains scenarios such as e.g. e-gov and e-health, in which the trusted management of identities and profiles is required. The present study provides a starting-point for further research and development of the IDN as the technological system underlying the e-Profile Management Systems towards a trust-enabling Web-wide service.

Keywords e-Profile · Identity · Trust · Infrastructure · Web of data

M. C. Pettenati (✉) · L. Ciofi · D. Parlanti · F. Pirri · D. Giuli
Electronics and Telecommunications Department, University of Florence, Florence, Italy
e-mail: mariachiara.pettenati@unifi.it

L. Ciofi
e-mail: lucia.ciofi@unifi.it

D. Parlanti
e-mail: david.parlanti@unifi.it

F. Pirri
e-mail: franco.pirri@unifi.it

D. Giuli
e-mail: dino.giuli@unifi.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_4,

4.1 Introduction

A user digital identity is a collection of personal data associated to a specific entity that could be a user or an institution etc. Digital identity (e-Identity) expressed by selected aggregates of personal data, intervenes very concretely in many facets of people's lives: their private life, their family life, their social life, their work life, their geographical mobility and the way they conduct business activities, their citizenship, their biological life, their life as a customer, etc. The digital profile, i.e. the identity associated with personal data, intervenes in three main contexts: (a) the access control based on identification to restricted resources or areas (b) the exploitation of identity information to provide (personalized) services the monitoring management of such processes to enable accountability [1].

The current technology evolutions, including Web 2.0, Cloud computing, the Linked Data, the Internet of Things and others still to come, will bring more personal data collection, a higher persistency of data in digital space, higher scales and more heterogeneity, pervasiveness and increased complexity. New developments in data mining and data fusion allow identities to be constructed from data that has not previously been considered identifying. Systems often do not control this kind of data as tightly as traditionally identifying data [2]. In this environment security, privacy and trust can be very difficult to monitor, verify and enforce [3, 4, 5].

Since this facts cause the explosion of the problem complexity, the need of a leap transition approach to digital profiles management is always more evident. Handling distributed granular identity/profile information via the Web, in such a way that it can ground the development of trusted e-Services, is considered a cornerstone of the Web Science Research Roadmap.[1]

In this paper we propose to adopt a research perspective aiming at embedding solutions at an infrastructural level targeting the issues related to *trust-enabling use of distributed data while coping with personal data management in the Web*. Our contribution in this direction is substantiated in the conceptual design and partial implementation of the InterDataNet (IDN) architecture allowing trust-enabling reuse and manipulation of interlinked information fragments on a Web-wide scale [6]. InterDataNet offers the infrastructural facilities to allow the trusted creation of IDN compliant application to support the management of digital profiles as basic global e-Service. The e-profile basic service is thus approached as a horizontal service handled by an *IDN Overlay Network of e-Profile Service Providers*, which can be made commonly shareable and accessible by heterogeneous end e-Services managed by third party Service Providers in a trust-enabling way, while guaranteeing consistence, availability and protection according to agreed policies of user personal information across several scenarios, contexts, and platforms.

[1] http://webscience.org

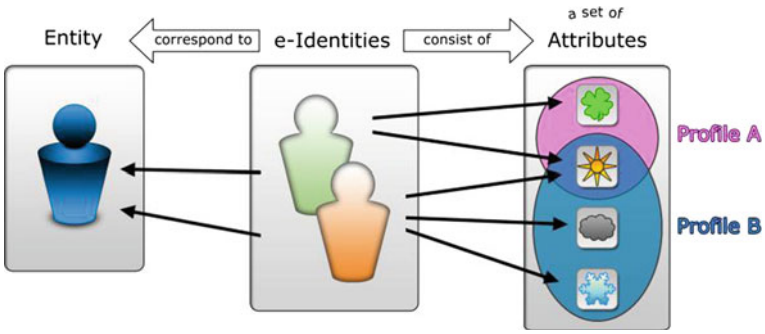


Fig. 4.1 Identity and profile

4.2 Terms and Definitions

4.2.1 Identity and Profile

A user digital identity is a collection of personal data associated to a specific user. A profile permits to store the description of the characteristics of person. This information can be exploited by systems taking into account the persons' characteristics and preferences, for instance to personalize interactions. Profiling is the process that refers to construction of a profile via the extraction from a set of data. Profiles are also very elaborated in online social networking services such as Facebook, Google profile, LinkedIn, in which people have the possibility to describe their identity attributes.

Profiles and entities are interrelated since identity is the connection between a profile, a set of attributes, and an entity (see Fig. 4.1). Some credentials (authentication) may be required from the user to access or create a profile. A user can have multiple identities as well as multiple profiles.

4.2.2 Trust and Trustworthiness

According to the RISEPTIS report [3] *trust* is defined as a three-part relation: *A trusts B to do X*, where A and B can be humans, organizations, machines, systems, services or virtual entities. Trust has the property of being: *context* dependent; contingent on *time;* influenced by *a number of variables* (such as reputation, recommendation, etc.); related to *identity and authentication information* (trust is easier to establish when such information about the third party are known).

Transposing the "classical" concept of trust in the digital space implies posing *trustworthiness* requirements on the system and service architecture.

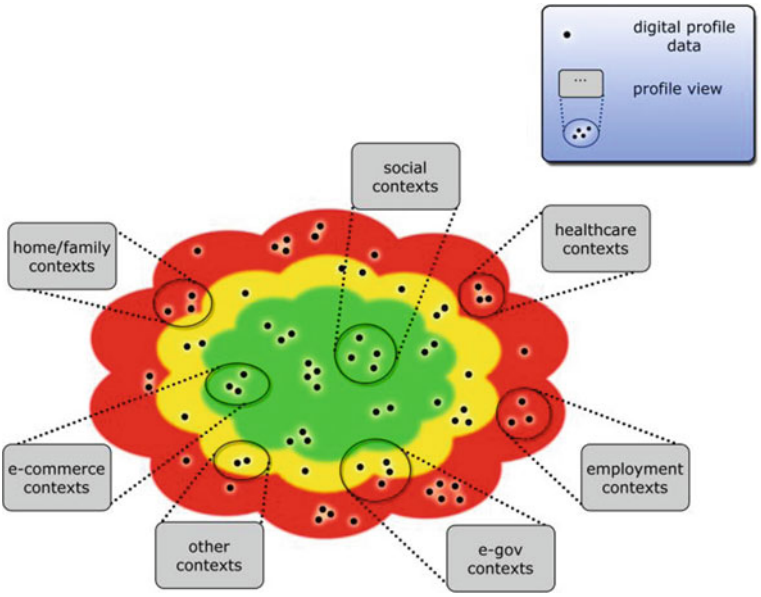


Fig. 4.2 User digital profile views according to different social contexts

Trustworthiness relates to the level of trust (is an attribute, a property) that can be *assigned by A to B to do something X in a given context*. Trustworthiness is not an absolute value and is context-dependent [3].

From the point of view of trust, a major weakness of the Internet comes from the lack of reliable verification of claims, including *identity* [7]. Consequently, mechanisms ensuring accountability [1], auditing, non-repudiation and law enforcement are increasingly difficult to implement [3].

Since at the basis *of trust lies the assessment of claims on the party to be trusted*, a basic framework for managing claim verification, including identity, non-repudiation, creditworthiness, reputation etc. is needed to develop federated, open and trustworthy platforms in various application sectors. This will allow the realization of concrete scenarios in which the user digital profiles are managed according to the different contexts (health care, government services, public procurement, commerce, etc.), as show in Fig. 4.2.

4.3 Web-Based Profile and Identity Management

The trustworthy management of identities and profiles on the Web is at the heart of the debate in many technical communities, especially in relation with the dominant phenomenon of social media [8].

Social media and networks are currently unsuitable for providing the required system properties of a global basic profiles management services because of three main issues:

1. The lack of granularity in the access and management of the single data element composing the user profile, thus limiting users data portability
2. The lack of control on *collaboration-oriented* attributes (e.g. privacy, licensing, security, provenance, consistency, versioning and availability) of the granular data;
3. The lack of infrastructural support to (1) and (2).

In current social media, users identity and profiles can easily be entered, but only accessed and manipulated via proprietary interfaces, so creating a "wall" around connections and personal data [9]. The lack of a truly universal, open, and distributed Web of data architecture deeply impacts the everyday experience of the Web of many users. Two main opposite forces drive current solutions and approaches [8]:

a. Data portability and linkability: An ordinary user can not download his/her own data and share and reuse it how he/she likes.
b. Privacy: A user cannot control how their information is viewed by others in different contexts by different applications even on the same social networking site, which raises privacy concerns. Privacy means giving people control over their data, empowering people to they can communicate the way they want. This control is lacking if configuring data sharing is effectively impossible or data disclosure by others about oneself cannot be prevented or undone.

The issue of granular data management is addressed by the current Web of Data approach [10] aimed at providing machine-readable data, given explicit semantics and published online, coupled with the ability to link data in distributed data sets. If this approach could seem to satisfy the property of granularity in access and management of information fragment, improvements are still sought in order to provide collaboration-enabling (i.e. privacy enhancing) capabilities [10]. However, current state of the art Web of Data approaches lead us back to a complex system solution in which above property (3) is not at all addressed let aside satisfied [11]. This introduces a higher level of complexity of the system on managing digital profiles at a global level than the one verified in the Web of Documents.

4.3.1 Identity Management Systems

Electronic Identity Management (eIdM) systems integrated in services provided by industry or by public administrations available at the state of the art [12–14] are often application-specific, centrally managed solutions designed with different assumptions and built with different goals in mind. As a result, they generally

lack seamless interoperability. Available technologies like OAuth [15], OpenID [16], WebID [17] are starting to provide viable solutions to opening up these silos [17].

OpenID centralises the authentication step at an identity provider, so that users can identify themselves with one site (an OpenID identity provider) and share their profile data with another site, the relying party. A user need only remember one globally unique identity. Once the OpenID provider is discovered, a shared secret is established in between the provider and the relying party, allowing them to share data.

WebID [17], uses Transport Layer Security (TLS) i.e. cryptographic protocols that provide security for communications over the Internet and client-side certificates for identification and authentication. However, certificate management and selection in browsers still has a lot of room for improvement on desktop browsers. Furthermore, by tying identity to a certificate in a browser, users are tied to the device on which their certificate was created. In fact a user profile can publish a number of keys for each browser, and certificates are cheap to create.

4.3.2 Profile Management Systems

A number of standards exist for profile and relationship information on the Web. One distinction among them is what data format the profile is in (plaintext, XML, RDFa) and whether or not they are easily extensible. Even more importantly, there are differences in how any particular application can try to discover and access the profile data and other capabilities that the digital identity may implement. OpenSocial [18] and PortableContacts [19] are two examples of profile management systems, dealing with the portability of portions of the profile, related to the users relationships and networks.

An increasingly popular profile standard is PortableContacts, which is derived from vCard, and is serialized as XML or, more commonly, JSON. It contains a vast amount of profile attributes, such as the "relationshipStatus" property, that map easily to common profiles on the Web like the Facebook Profile. More than a profile standard, the PortableContacts profile scheme is designed to give users a secure way to permit applications to access their contacts, depending on XRDS for the discovery of PortableContact end-points and OAuth for delegated authorization. It provides a common access pattern and contact scheme as well as authentication and authorization requirements for access to private contact information.

OpenSocial is a collection of Javascript APIs, controlled by the OpenSocial Foundation, that allow Google Gadgets (a proprietary portable Javascript application) to access profile data, as well as other necessary tasks such as persistence and data exchange. It allows developers to easily embed social data into their Gadgets. The profile data it uses is a superset of the PortableContacts and vCard 3.0 data formats.

OAuth (Open Authorization) is an IETF standard which lets users share their private resources on a resource-hosting site with another third-party site without having to give the third-party their credentials for the site and so access to all their personal data on the social site. This standard essentially defeats the dangerous practice of many early social networking sites of accessing for the username and password of an e-mail account in order to populate a list of friends. OAuth is not really related to identity, rather it is about data authorization and API access to data, which necessarily is an important part of the profile management.

4.3.3 Use of Identities and Profiles in Contexts

Identity, Profiles and their management are also at the heart of European Community initiatives, aiming at tracing directives for the research and future developments in this field. With the respect, three important activities founded by European Community are worth mentioning: FIDIS and PRIME and STORK [20–22]. Instead starting from the assumption that the identity of a person comprises many partial identities of which each represents the person in a specific context or role, there projects hinge around the idea of choosing and developing appropriate partial identities with respect to the current application needs.

FIDIS is an international interdisciplinary Network of Excellence on the Future of ID-entity in the Information Society (2004–2009) and its concern has been mainly focused in the realization of a set of recommendations and specifications about the Identity topic. One of the objects of investigation for the FIDIS research community is the interoperability of identity management systems from the technical, policy, legal and socio-cultural perspectives.

The PRIME project (2004–2008) under the European Union's Sixth Framework Programme aimed to demonstrate the viability of privacy-enhancing identity management. The guiding principle in the PRIME project is to put individuals in control of their personal data within the scenario introduced a limited application domain, on-line shopping.

The STORK project (2009–2011), instead, aims to establish an European e-ID Interoperability Platform that will allow citizens to establish new e-relations across borders, just by presenting their national e-ID. STORK will test pragmatic eID interoperability solutions, implementing several pilot cross-border eID services chosen for their high impact on everyday life as the scenario named Student Mobility or the scenario Change of Address, to assist people moving across EU borders. Since the latter service has a great impact on the mobility of citizens, it has been included as one of the 20 basic services within the i2010 EU Commission eGOV Action Plan.

To the extent of lowering system complexity and increase the manageability of the problem, we propose the introduction of IDN, a common infrastructure which can provide e-Services Providers the infrastructural facilities to develop

e-profile management applications—i.e. IDN-compliant applications—deployable at a Web-wide scale.

Our solution introduces the capability of handling distributed data (identities and profile attributes) structured into documents (profile views) and provide collaboration-oriented management of granular elements (e.g. disclosing profile views personalising policies on profile elements/attributes related to privacy, security, usability and accessibility, self-management, role-based access/functions, transparency, licensing, etc. [12, 23]).

4.4 InterDataNet Middleware Infrastructure

The approach proposed in this paper builds on the introduction of granular addressability of data and offers a uniform Web-based read-write interface to distributed heterogeneous profile data management in a trust enabling environment. This approach is sustained by the IDN [6] middleware conceptual infrastructural solution.

InterDataNet infrastructure provides distributed structured data management and provides collaboration-oriented functions, offering a document-centric overlay infrastructural approach to e-profile management.

InterDataNet is focused on the realization of a layered infrastructure in which:

- Information is made available and managed with arbitrary granularity.
- Information is uniformly managed (CRUD) with a uniform interface.
- Authentication and authorization services are provided.
- Collaboration enabling mechanism, namely privacy, licensing, security, provenance, consistency, versioning and availability.
- Interfaces to legacy sources are provided.

IDN framework is described through the ensemble of concepts, models and technologies pertaining to the following two views: *(1) IDN-IM (InterDataNet Information Model)*. It is the shared information model representing a generic document model which is independent from specific contexts and technologies; *(2) IDN-SA (InterDataNet Service Architecture)*. It is the architectural layered model handling IDN-IM documents (it manages the IDN-IM concrete instances allowing the users to "act" on pieces of information and documents).

The IDN-SA exposes an *IDN-API (Application Programming Interface)* on top of which IDN-compliant Applications can be developed (see Fig. 4.3). IDN is not the e-Identity/e-Profile management system, rather it is an infrastructure allowing the Web based distributed collaborative management of elementary information fragments, i.e. user profiles elements and attributes, through a specifically developed IDN-compliant Application (Fig. 4.4).

IDN Information Model (IDN-IM) is the graph-based data model to describe interlinked data representing a generic document model in IDN, independently from specific contexts and technologies. It defines the requirements, desirable

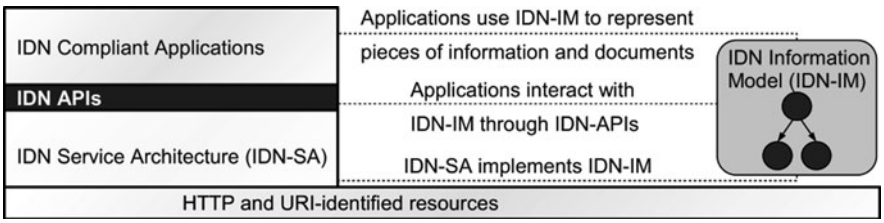


Fig. 4.3 The IDN framework

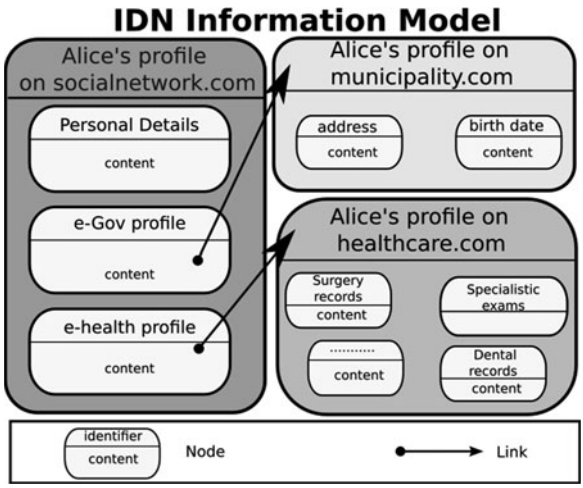


Fig. 4.4 The IDN information model

properties, principles and structure of the document as it is seen by IDN-compliant applications and it is the starting point from which it has been derived the design of IDN architecture. Generic information modelled in IDN-IM is formalized as an aggregation of data units. Each data unit is assigned at least with a global identifier and contains generic data and metadata; at a formal level, such data unit is a node in a directed acyclic graph (DAG). The abstract data structure adopted within IDN-environment to handle generic data units is named IDN-Nodes. An IDN-document structures data units and it is composed by nodes related to each other through "links". IDN-documents can be inter-linked.

The second element of the IDN framework is the IDN-Service Architecture, that is layered in such a way to manage different levels of abstraction of information.

IDN Service Architecture (IDN-SA)—Fig. 4.5 is made up of four layers: Storage Interface, Replica Management, Information History and Virtual Resource. Each layer interacts only with its upper and lower level but also relies on the services offered by IDN naming system. IDN-Nodes are the information that the layers exchange in their communications. It has to be highlighted that in each layer a different type of IDN-Node is used: SI-Node, RM-Node, IH-Node and VR-Node.

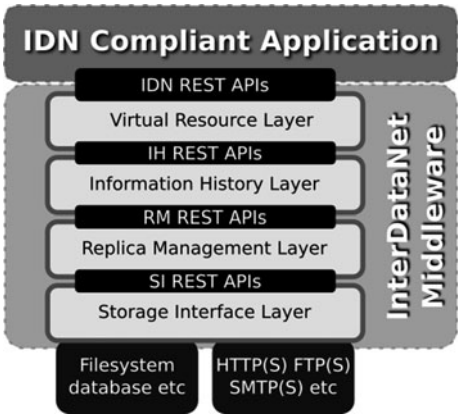


Fig. 4.5 IDN-service architecture

Each layer has as input a specific type of IDN-Node and applies on it a transformation on the relevant metadata to obtain its own IDN-Node type. The transformation (adding, modifying, updating and deleting metadata) recalls the encapsulation process used in the TCP/IP protocol stack. The different types of IDN-Nodes have different classes of HTTP-URI as identifiers.

1. *Storage Interface* (*SI*) provides the services related to the physical storage of information and an interface towards legacy systems. It provides a REST-like uniform view over distributed data through URL names. It provides create, read, update and delete functions of URL-addressed resources.
2. *Replica Management* (*RM*) provides a delocalized view of the resources to the upper layer.
3. *Information History* (*IH*) manages temporal changes of information.
4. *Virtual Repository* (*VR*) manages the document structure.

The communication between IDN-SA layers (see Fig. 4.5) follows the REST paradigm i.e. it is based on common HTTP messages containing a generic IDN-Node in the message body and IDN-Node identifier represented by an HTTP-URI in the message header. IDN-Nodes are identified by HTTP-URI as identifiers, thus providing substantial compliancy with Linked Data and Open Data initiatives. IDN-SA is implemented using a set of CGI program modules running into standard HTTP server and offers a RESTful interface. The interaction between IDN-compliant applications and IDN-SA follows the HTTP protocol as defined in REST architectural style. CRUD operations on application-side will be enabled for the manipulation of data on a global scale within the Web. As a consequence of the adoption of the HTTP-URI paradigm for Nodes identification, each IDN-Node can be under a different authoritative server, thus allowing trust-enabling data cooperation strategies.

It is worth pointing out that anything can be an IDN document because the IDN information model provides a general method to represent information. The case addressed in this presentation is the e-profile (or e-identity) but we can envisage to

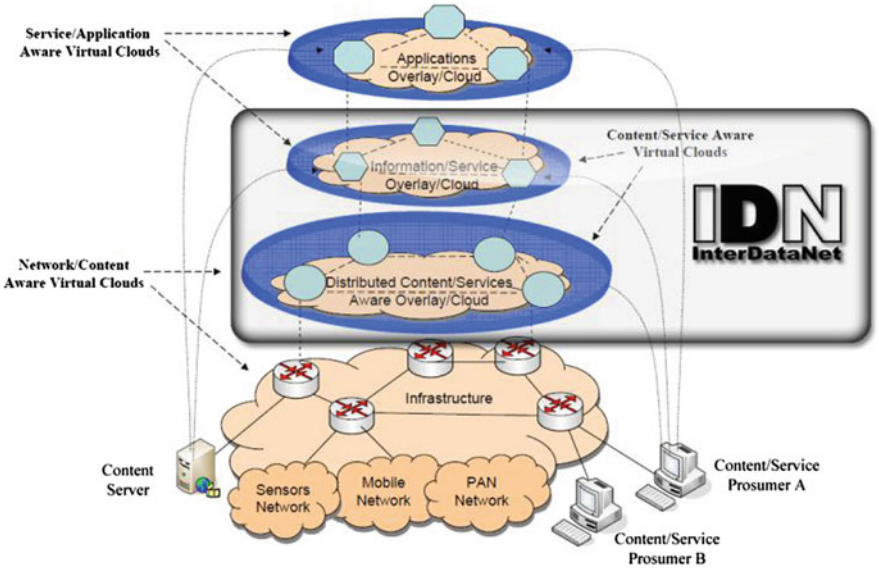


Fig. 4.6 InterDataNet Overlay Network (adapted from Zachariadis [24])

manage through the IDN architecture other kinds of IDM-IM compliant documents, such as—for instance—trust policy and Security policy documents, just to mention some.

At last it is worth saying also that IDN approach enables collaboration because it allows creating a "collaborative space" defined by the IDN-Information Model properties and rules. In such space users can share contents/information/documents formalized with respect to this model and work around them. The collaborative process is made concrete through the creation of documents endowed with arbitrarily complex life cycles evolving as a consequence of the users' interactions. The IDN-Information Model life cycle management is supported at an Application level because it is strictly application-dependent, however, it can leverage on a set of collaboration-enabling capabilities which are, in this case, provided by IDN-Service Architecture, this property can be addressed with the following terms: privacy, licensing, security, provenance, consistency, versioning and availability.

4.5 InterDataNet Overlay Network

InterDataNet provides a content-centric abstraction level hinging on the IDN-Information Model and related handling mechanism (the IDN Service Architecture) to provide enhanced content/information-centric services to Applications, such as those handling the e-Profiles, as illustrated in Fig. 4.6.

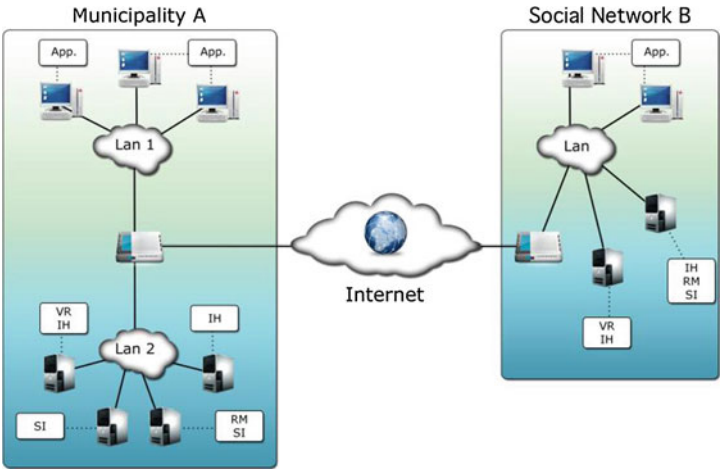


Fig. 4.7 Possible implementation of the IDN Overlay Network

InterDataNet proposes a distributed data integration approach which would act as a de-facto Overlay Network on the underlying integrated systems. A possible implementation of the Overlay Network is schematised in Fig. 4.7, in which two organizations responsible of managing personal data of the users are connected via the IDN Overlay: the Municipality A and Social Network B.

The layers of the IDN-SA are implemented in specific HTTP servers, networked inside interoperating organizations which communicate via the Internet. Each HTTP server may implement only some layers of the IDN-SA stack, as it is the case shown in Fig. 4.7.

The IDN-compliant Application built on top of the Overlay Network is entitled to the profiles management, i.e. the aggregation of (open and private) data from different users' e-identity sources (e.g. Content Providers, Social Networks, Context-information Providers, Public Administrations, etc.). Under the user's permission, the Service Provider provides the user's profiles views to third party Service Providers for the creation of personalized services. The Overlay Network can be used as a virtual "repository" providing a common representation for profile management through a uniform API. When considered as a repository, the Overlay could offer direct querying of information to interested/authorized third parties, while also allowing exploitation of structured profiles by the overlay nodes themselves. The implementation of the integration Overlay Network allows both an internal interaction and an external interaction. The internal interaction directly targets nodes of the overlay structure and it is based on the local exploitation of the global overlay capabilities for easing development of data-centric applications while the external interaction targets commercial third-parties interested in the development of applications e-identity management. Systems participating to the Overlay are not forced to modify their internal interaction model due to their integration as overlay nodes.

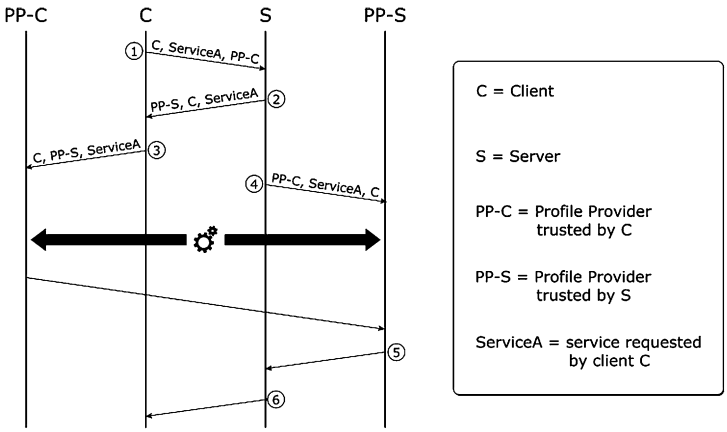


Fig. 4.8 Sequence diagram of the usage scenario

4.5.1 Usage Scenario of the IDN Overlay for the e-Profile Management

IDN Overlay envisions a trustworthy exchange of views over personal data thanks to a network of trusted Profile Providers as mean to create a global e-Profile Management system. In this vision each user is free of choosing the preferred Profile Provider in order to manage her own personal data as IDN documents modelled according to the IDN Information Model. The user trusts her Profile Provider which will handle in a secure way his personal data and guarantee other known Profile Providers the personal data trustworthiness. Profile Providers are known when they have established a certain level of agreement.

We now consider an example of a bank transfer scenario where the flow of money is guaranteed thanks to an agreement between banks. In this analogy the book represents the service, the money represents the required authorization(s) and the banks represent the trust providers. Even if customers are not aware of agreement details, they normally trust their bank to behave correctly upon their request. The considered scenario is referred to the following use case: a customer (C) wants to buy a book from an online shop (S). In this scenario the money transfer between the customer's bank and the online shop bank represents the trust chain among different Profile Providers. The trust chain is established whenever a client wanting to access a service is required to disclose some personal data to the Service Provider and to assure that the data provided are trustworthy.

The steps of the scenario are the following, (illustrated also in Figs. 4.8 and 4.9):

Step 1: C sends S a request to buy a book and declares that she will pay with a bank transfer, where the bank is PP-C;

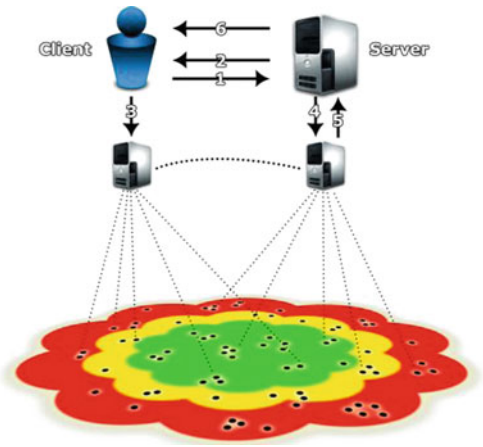


Fig. 4.9 Main functions of the e-profile Overlay Network

Step 2: S says to C to dispose for a bank transfer on a specific bank account number in the bank PP-S;
Step 3: C grants PP-C the right to transfer money from her personal account to the specified bank account at PP-S;
Step 4: S request PP-S to receive money from PP-C;
…
PP-S trusts PP-C following a chain of trusted PP-X;
PP-S receives money from PP-C;
…
Step 5: PP-S notifies S the bank transfer accomplishment from PP-C;
Step 6: S notifies C the successful completion of the transaction.

PP-S can receive money from PP-C either because banks share agreements or because they trust each other thanks to a kind of trusted chain of PP-X. IDN allows PP-C and PP-S to communicate in terms of documents IDN-IM. The realization of a chain of trusted PP-X that allows them to trust what it is contained in the IDN-IM documents, and is known in literature as *brokered trust* [25].

4.6 Conclusions

In this paper we discussed some architectural issues related to the management of distributed users identities profile on the Web. Being the notion of digital identity and digital profile strictly interrelated, we reviewed some of the main current approaches for the Identity and Profile Management, identifying two main opposing forces: *personal data portability* and *privacy* and highlighting that the trade off between those two opposite forces leads to technical and organizational solutions partially addressing the overall system problem. We therefore identified the need of a paradigm shift in the underlying architecture, capable of addressing

the need of granular distributed data management over the Web while at the same time supporting privacy-enabling feature at an infrastructural level. Our proposal in this direction is the IDN framework, a distributed layered architecture providing information fragment management in a document-centric perspective and leveraging on a set of collaboration-enabling properties, namely privacy, licensing, security, provenance, consistency, versioning and availability.

In the approach presented in this paper the proposed architecture offers its functionality to IDN-compliant applications built on top of it, eventually allowing the creation and management of a global e-profile Overlay Network spanning multiple administrative domains. This conceptual proposal is described through a scenario highlighting how seamless inter(data)working offered by the architecture facilitates the management and disclosure of e-profile views between systems.

At present, our research on IDN is evolving both as a project for specific case studies as well as a vision. In the former line, its implementation and studies are related to the domain of e-Government services, i.e. change of address, which is focused at proving both the viability and applicability of the technical and organisational solutions proposed; in the latter evolutionary line the IDN team seeks a continuous fine tuning to reach the most possible compliancy with the Linked Data approaches in order to maximise reciprocal potential impact.

Several open issues are still to be addressed in order to concretise the specific proposed approach of the Overlay Network for e-profile management. Our current efforts aim to integrate the approaches of two well-known technologies as OAuth and OpenID in the brokered trust model.

The research challenges addressed by this work are in line with the W3C recommendations [26] and can be valued within the current European initiatives [27, 28] of development of a common framework for federation and interoperability of governmental profile Management systems that can form the basis of a wide digital claim management framework, compliant with the legal framework for data protection and privacy.

Acknowledgments We would like to acknowledge the precious discussion with Dr. G. Vannuccini. We are also thankful to Mr. R. Billero, S. Turchi and L. Capannesi for the technical support in the implementation IDN-Service Architecture.

References

1. Weitzner, D.J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., Sussman, G.J.: Information accountability. Commun. ACM **51**(6), 82–87 (2008). doi:10.1145/1349026.1349043
2. Skillicorn, D., Hussain, M.: Personas: beyond identity protection by information control (a report to the privacy commissioner of Canada). http://research.cs.queensu.ca/~skill/opccreport.pdf (2009)
3. Research and Innovation for Security, Privacy and Trustworthiness in the Information Society (Riseptis) Advisory Board: Trust in the information society. http://www.think-trust.eu/riseptis.html (2009)

4. Burdon, M.: Commercializing public sector information privacy and security concerns. Technol. Soc. Mag. IEEE **28**(1), 34–40 (2009)
5. Hoikkanen, A., Bacigalupo, M., Compañó, R., Lusoli, W., Maghiros, I.: New challenges and possible policy options for the regulation of electronic identity. J. Int. Commer. Law Technol. **5**(1), 1–10 (2010)
6. Pettenati, M.C., Innocenti, S., Chini, D., et al.: Interdatanet: a data web foundation for the semantic web vision. IADIS Int. J. WWW/Internet **6**(2), 16–30 (2008)
7. Hoffman, R.R., Lee, J.D., Woods, D.D., Shadbolt, N., Miller, J., Bradshaw, J.M.: The dynamics of trust in cyberdomains. Intell. Syst. IEEE **24**(6), 5–11 (2009)
8. W3C Social Web Incubator Group: Final report. http://www.w3.org/2005/Incubator/socialweb/wiki/FinalReport#The_Problem_of_Walled_Gardens (2010)
9. Berners-Lee, T.: Socially aware cloud storage. http://www.w3.org/DesignIssues/CloudStorage.html (2009)
10. Bizer, C., Heath, T., Berners Lee, S.T.: Linked data—the story so far. Int. J. Semant. Web Inf. Syst. Spec. Issue Linked Data **53**(3), 1–22 (2009)
11. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic web. Presented at Linked Data on the Web Workshop (LDOW2010) at WWW'2010 (2010)
12. Pacyna, P., Rutkowski, A., Sarma, A., Takahashi, K.: Trusted identity for all: toward interoperable trusted identity management systems. Computer **42**(5), 30–32 (2009)
13. Chadwick, D.W., Inman, G.: Attribute aggregation in federated identity management. Computer **42**(5), 33–40 (2009)
14. Fioravanti, F., Nardelli, E.: Identity management for e-government services. In: Chen, H., Brandt, L., Gregg, V., Traunmüller, R., Dawes, S., Hovy, E., Macintosh, A., et al. (eds.) Digital Government: E-Government Research, Case Studies, and Implementation, pp. 331–332. Springer (2008)
15. OAuth: http://oauth.net/
16. OpenID Foundation: http://openid.net/
17. WebID: ESW Wiki. http://esw.w3.org/WebID
18. OpenSocial: http://www.opensocial.org/
19. Portable Contacts: http://portablecontacts.net/
20. FIDISCoord (DR), FIDISCoord (DR): Future of IDentity in the information society. NoE (network of excellence). http://www.fidis.net/
21. PRIME—Privacy and Identity Management for Europe: https://www.prime-project.eu/
22. STORK—Secure Identity across Borders Linked: https://www.eid-stork.eu/
23. Giuli, D., Pettenati, M.C., Parlanti, D.: Individual and social needs motivating trust-enabling intermediation services. In: Proceedings of the 13th European conference on cognitive ergonomics: trust and control in complex socio-technical systems. Zurich, Switzerland, pp. 111–112 (2006)
24. Zahariadis, T., Daras, P., Bouwen, J., Niebert, N., Griffin, D., Alvarez, F., Camarillo, G.: Towards a content-centric internet. In: Tselentis, G., Galis, A., Gavras, A., Krco, S., Lotz, V., Simperl, E., Stiller, B., Zahariadis, T. (eds.) Towards the Future Internet—Emerging Trends from European Research, pp. 227–236. IOS Press, Amsterdam (2010)
25. Oasis Trust Models Guide Lines (Draft Document): http://www.oasis-open.org/committees/download.php/6158/sstc-saml-trustmodels-2.0-draft-01.pdf
26. Improving Access to Government through Better Use of the Web: W3C interest group note 12 May (2009)
27. e-Infrastructure Reflection Group: e-IRG roadmap 2010. http://www.e-irg.eu/publications/roadmap.html (2010)
28. The European Interoperability Framework Recommendations. http://ec.europa.eu/idabc/en/document/3473/5887.html

Part II
Security, Energy Efficiency, Resilience and Privacy

Chapter 5
Context Privacy in the Internet of Things

Laura Galluccio, Alessandro Leonardi, Giacomo Morabito and Sergio Palazzo

Abstract Recent advances in electronics, communications, and information technologies make possible to collect, process and store a large amount of information from the physical world. Once generated, such information is rarely erased and it is almost impossible to fully control its diffusion. This raises serious privacy problems. In this paper attention is focused on the privacy problems occurring in the pervasive and heterogeneous scenarios characterizing the Internet of Things. Since access to the data content can be prevented utilizing traditional security mechanisms, focus will be put on context privacy, which concerns the protection of sensible information that requires to be guaranteed even if data content is fully secured. In this paper, the privacy problems will be described and analyzed along with the main solutions that have been proposed so far in the literature.

Keywords Sensor network · Privacy

This work has been partially supported by the Italian National Project: Wireless multiplatfOrm mimo active access netwoRks for QoS-demanding muLtimedia Delivery (WORLD), under grant number 2007R989S.

L. Galluccio · A. Leonardi (✉) · G. Morabito · S. Palazzo
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni,
University of Catania, Catania, Italy
e-mail: alessandro.leonardi@diit.unict.it

L. Galluccio
e-mail: laura.galluccio@diit.unict.it

G. Morabito
e-mail: giacomo.morabito@diit.unict.it

S. Palazzo
e-mail: sergio.palazzo@diit.unict.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_5,

5.1 Introduction

Over 2500 years ago Aristoteles set a distinction between *public* and *private* life. This is the first documented discussion related to the problem of *privacy* protection. Since then, privacy has always been recognized as a crucial requirement for the development and free expression of the personality. Accordingly, privacy is enumerated among the fundamental human rights in the *Universal Declaration of Human Rights* and its protection is regulated in the legislation of all developed countries. However, legislation efforts in favor of the protection of privacy are contrasted by law enforcement requirements that impose the disclosure of all possible information about an individual if this is needed for security reasons.

As information and communication technologies evolve, new privacy problems arise. Accordingly, the need for appropriate legislation has been highlighted by Warren and Brandeis as a conclusion of their analysis of the impact of communication and media technology. Warren and Brandeis completed their analysis in 1890 and were worried because of the increase in threats to privacy due to the diffusion of newspapers and photographs.

In the recent past, new communication technologies and media have been introduced, that make it possible to deploy computing and communication devices in most of the objects we use in our daily life. It follows that actual realization of the ubiquitous computing concept introduced in the Mark Weiser's visionary paper [20] is close to our reach. Ubiquitous computing involves cooperation between objects, which requires "things" to communicate between themselves as nodes of a world-wide IPv6-based network which includes today's Internet and is called the *Internet of Things*[1] (IoT). Actual deployment of the IoT has a possible impact on our privacy that goes far beyond what Warren and Brandeis were fearing at the end of the nineteenth century. Accordingly, in the last few years a large research effort has been devoted to identify the relationships between the technologies involved by the IoT paradigm and privacy. Indeed, it is evident that privacy must be considered as a *quality of service* (QoS) parameter, exactly like loss probability and delay.

In this paper, we analyze the privacy problems and related solutions in the context of the technologies involved by the IoT, that is wireless sensor, RFID, and ad hoc networks which will be key components of the ubiquitous computing infrastructures envisioned in the IoT.

More specifically, for worth of classification, in Sect. 5.2 we provide a taxonomy of privacy, while in Sect. 5.3, which is the focus of our work, i.e., the context privacy, we survey the most relevant research results in the context of the technologies which will play crucial roles in the IoT, that is of wireless sensor, RFID and ad hoc networks. Finally, in Sect. 5.4 some conclusions are drawn.

[1] For a general description of the Internet of Things concepts, technologies, and open research issues refer to [3].

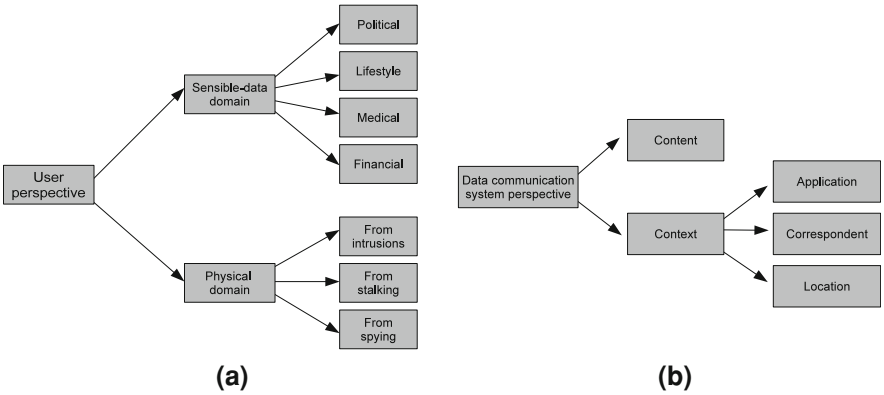


Fig. 5.1 (**a**) Classification of privacy from the perspective of a user (**a**) and a data communication system (**b**)

5.2 Taxonomy

In the years, several definitions of privacy have been proposed. A good one can be found in [13]: *Privacy* is the condition of not having *undocumented* personal information known or possessed by others. Here, personal information is considered *documented* if it belongs to public records, that is, newspapers, court records, or other public documents. Many types of privacy can be distinguished. In Fig. 5.1a, we outline a classification of the different types of privacy.

From a user perspective, we can consider:

- *Physical domain privacy*, concerning the defense from intrusions into one's physical space. It involves protection from intrusions into private properties (i.e., *privacy from intrusions*), from being stalked (i.e. *privacy from stalking*), and from being unconsciously observed and/or heard (i.e. *privacy from spying*).
- *Sensible-data domain privacy*, concerning the protection of information about the individual that she/he does not want to be public such as political preferences, lifestyle (e.g., family habits, sexual preferences, etc.), health-related information, and financial information.

Initially, communication and information technologies were regarded as a threat for Sensible-data domain privacy only. Now that such technologies are integrated in objects that we use during our daily activities and that such objects can be reached with the same simplicity of other Internet nodes, they can also impact Physical privacy. As an example, a multimedia sensor network integrating cameras [1] allows to spy and even stalk unaware people. Furthermore, stalking a person is also possible by tracking the positions of the RFIDs deployed in objects that are normally carried by such person.

As shown Fig. 5.1b, from a data communication system perspective, we can distinguish:

- *Content privacy*, concerning the protection of the content of the data being exchanged by communication devices.
- *Context privacy*, concerning the protection of other sensible information that can be inferred even when the message content is secured. Within Context privacy, which is the focus of this paper, we can further distinguish:
  - *Application privacy*, concerning the protection of the telematic application being run. Observe that such information can be inferred from the socket port being utilized as well as the data trace, e.g., statistics of the packet size and the time interval between transmissions [19].
  - *Correspondent privacy*, concerning the protection of the identity of the other peer in a communication. This can be easily inferred by the IP address of the packets being exchanged [14].
  - *Location privacy*, concerning the protection of the user location information. In the perspective of location privacy, we can also consider the *itinerary privacy*, which identifies the ability to protect nodes' mobility patterns.

Over the past years, computer security has focused on the content privacy, that is, on protecting the content of data being exchanged between communicating nodes, thus ensuring data confidentiality, integrity, and availability. However, even if such objectives were reached and, therefore, content privacy was fully achieved, context privacy would still remain a problem. Accordingly, in the following we will disregard content privacy and focus on context privacy only.

5.3 Context Privacy

Threats to context privacy stem from the ability of adversaries nodes to gather information from observation of users' communications without direct access to the contents of the messages exchanged.

In the following we will focus on the context privacy issues that are particularly critical in the IoT scenarios. More specifically, we will analyze the threats to context privacy along with the solutions proposed to deal with them. In fact, we will address application privacy in Sect. 5.3.1, correspondent privacy in Sect. 5.3.2, and location privacy in Sect. 5.3.3

5.3.1 Application Privacy

In IoT scenarios objects will cooperate with each others by running appropriate web-based applications [18]. Accordingly, violating application privacy means gathering information about the objects that are being utilized and the type of use of such objects that is taking place.

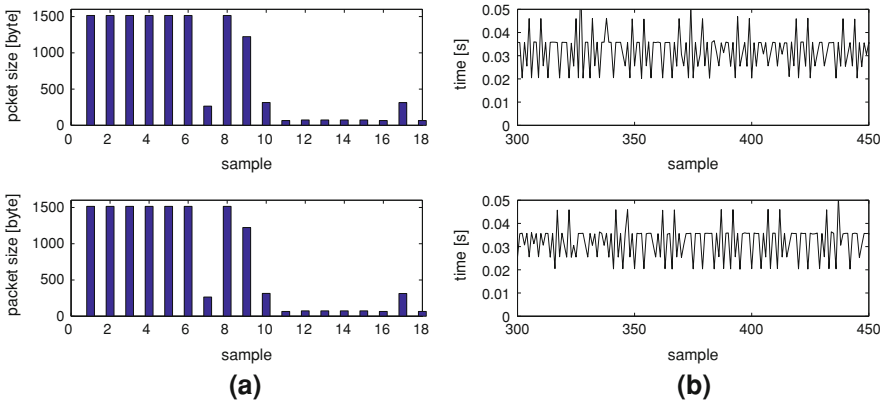


Fig. 5.2 Size of the packets received due to the download of the web home page of the School of Engineering of the University of Catania - http://www.ing.unict.it (**a**), inter-arrival times of the packets generated by a Skype conversation (**b**)

The application run by a node can be easily inferred in several ways. For example, the port number reported in the transport layer header can be analyzed to distinguish between different applications. Alternatively, the application being utilized can be inferred by observing the information reported in the source-destination addresses fields of the packets being exchanged (which will be dealt with in Sect. 5.3.2). Here we focus on another type of application privacy attack that can be accomplished through traffic analysis.

Each application involves the exchange of a certain number of messages each one with a given size, so producing the so called *fingerprint* of the service. Fingerprints can be taken through passive sniffing of the traffic generated by and/or directed to the attacked node.

For example, given that there are no IoT application that have been actually implemented so far, in Fig. 5.2a we show the traffic traces, i.e., the sizes of the packets, received by running a traditional Internet service, i.e., downloading the home page of the website of the School of Engineering of the University of Catania (http://www.ing.unict.it) in two different time instants. The similarity between the two traffic patterns is evident. Therefore, it is possible to infer that a node is running a certain application with known fingerprint by analyzing the traffic trace with no need of accessing the packet content. The more messages in a fingerprint, the higher the chance that the fingerprint will be unique.

In Fig. 5.2b, instead, we show the interarrival times of packets generated by another traditional Internet application, that is, Skype in two different conversations. Observe that also in this case the similarities between the two traces are evident and therefore, it is quite easy to infer if a certain user is involved in a Skype conversation or not.

Traffic analysis attacks are possible in any communication scenario but can be carried out more easily when the wireless multihop communication paradigm is

applied, such as it is common in Internet of Things scenarios. In fact, the broadcast nature of the wireless medium makes eavesdropping simple and the multihop communication paradigm increases the chances to use it.

In literature, among various approaches, also *traffic shaping* has been proposed to increase application privacy. The obvious idea is to force all traffic flows to fit in a certain traffic pattern. In such a way all applications produce traffic traces with the same pattern and therefore, it is impossible to distinguish one from another through traffic analysis. As an example, we may think that traffic is shaped so as to make packets have constant size, S, and be transmitted at constant intervals, T.

However, it is well known that traffic shaping involves increase in the delivery delay and/or decrease in efficiency. Indeed, if the interval duration T is too low, then packet transmissions must be forced even if not needed, which implies a decrease in efficiency. On the contrary, if the interval duration T is too long, delay may increase which results in the impossibility to support realtime applications. Concerning the packet size S, if this is too low, then packet fragmentations may be frequent, which results in several efficiency problems. In fact: the number of accesses to the wireless medium increases, and this results in lower utilization of the wireless resources, as well as an increase in delay; the overhead increases as headers have to be included in each fragment of the packet.

Similar problems arise if the value of S is too high. This calls for an appropriate setting of the parameters T and S in order to minimize the performance degradation introduced by traffic shaping.

5.3.2 Correspondent Privacy

It is obvious that correspondent privacy is crucial in several IoT scenarios as it concerns protection of information that can be easily utilized to infer relationships between users and/or their favorite services, things, and sources of information.

Correspondent privacy can be attacked by tracing packets exchanged by corresponding nodes, which is quite simple in case of wireless multihop communications. Solutions aimed at protecting this type of privacy must achieve the so called *route un-traceability*. Several of them have been proposed to this purpose in the recent past. The most known are *onion routing* [8], ANODR [9] and multi-path routing [5], which will be described in the following of this section.

In *onion routing*-like protocols [8], special nodes, called *onion nodes*, which have public keys known by all nodes, are deployed in the network. A source S, which has a packet to be delivered to the destination D, randomly chooses a sequence of N of them which we denote $O_{i_1}, O_{i_2}, \ldots, O_{i_N}$. Then it prepares the packet and sets the *Destination_Address* field in the packet header equal to $\text{Addr}(D)$, where $\text{Addr}(x)$ represents the address of node x. This packet is encrypted using the public key of the onion node O_{i_N}. The resulting sequence of bits is inserted in the data field of a new packet in which the *Destination_Address* field is set equal to $\text{Addr}(O_{i_N})$. The resulting packet is encrypted using the public key of

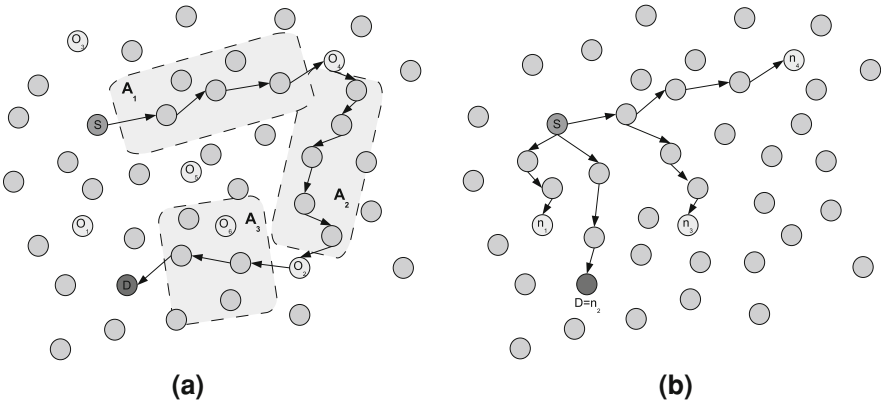


Fig. 5.3 Onion routing for route un-traceability (**a**), multi-path routing for route un-traceability (**b**)

$O_{i_{N-1}}$ and inserted in the data field of a new packet which is destined to $\text{Addr}(O_{i_{N-1}})$. The same procedure is repeated for $O_{i_{N-2}}, O_{i_{N-3}}, \ldots, O_{i_1}$. Finally, the packet is transmitted and forwarded through the network using a mechanism similar to the one utilized in IP packet encapsulation.

As an example, in Fig. 5.3a we show a case in which six onion nodes are deployed, i.e., $O_1, \ldots, O_6$, and the source S chooses a sequence of $N = 2$ onion nodes, that is, O_4 and O_2. Observe that using onion-routing, the addresses of the source S and destination D cannot be put in a relationship by eavesdroppers in the same area (i.e. $\text{A}_1, \text{A}_2, \text{A}_3$) and thus, onion routing solutions guarantee route un-traceability. Unfortunately, this is achieved at the expenses of an increase in the number of hops required to deliver the packet to the final destination, which causes an increase in the delay and higher consumption of both bandwidth and energy.

ANODR [9] is a solution that guarantees route un-traceability by means of route pseudonymity. A source S that wants to establish a connection with a destination D starts a route discovery phase in which a *route request* (RREQ) message encrypted using the public key of D is flooded into the network. In order to guarantee route un-traceability, at each hop a *trapdoor* function[2] is applied to the RREQ before it is broadcast. The trapdoor information utilized by the node is stored in the node itself (trapdoor functions are used instead of cryptographic functions because they are computationally simpler). Upon receiving the RREQ, the destination D sends back a *route reply* (RREP) message which traverses the shortest path between the source S and the destination D. Note that none of the nodes is aware of the paths between the source S and destination D, not even S and D. Furthermore, observe that

[2] A trapdoor function is a function that is easy to compute in one direction, and difficult to compute in the opposite direction (e.g., by finding its inverse) without some information, called the *trapdoor*.

ANODR is based on a reactive routing approach, accordingly topological information will not be exchanged, which results in increased location privacy too.

Finally, route un-traceability can be achieved by means of multi-path routing, as proposed in [5]. In such a type of solutions, in order to reach a certain destination D, a packet is forwarded by the source S along several, say M paths. One of such path will pass through the destination D. As an example, in Fig. 5.3b the packet is forwarded along 4 paths terminating in n_1, n_2, n_3, and n_4, respectively, where n_2 is the destination. In this way the attacker cannot identify the actual destination. Obviously, privacy increases as the value of M increases. However, such a privacy improvement is obtained at the cost of higher energy and bandwidth consumption, which calls for appropriate tradeoffs.

5.3.3 Location Privacy

Wireless multihop networks represent a challenging environment for the protection of location privacy because nodes have to distribute information about their position for topology management and routing purposes. Such messages can be easily eavesdropped by attackers who can get information about users' (or assets') location accordingly. To overcome this problem we can distinguish two different types of approaches:

1. Using pseudonyms so that the topology information can be exchanged without disclosing the identity of the specific users (or nodes) so as to prevent eavesdroppers to infer links between users and their position
2. Using routing protocols that do not require any exchange of topology information.

Regarding solutions of the first type, note that if each node is identified by means of one or more pseudonyms, then it can distribute information about its current position while protecting location privacy. In fact, relationship between user's identity and location can be disclosed by public security operators only. However, if the same pseudonym (or the same set of pseudonyms) is used for a long time, then this property does not hold anymore and, thus, appropriate solutions are required that allow nodes to change their pseudonyms over time.

One of such solutions is the *random node identification* (RNI) protocol proposed in [5]. Nodes can generate a new random pseudonym at any time, and flood it like an AODV HELLO message. In this way, nodes know each other by means of their pseudonyms that are used for communication and routing purposes. Such solutions protect privacy but have serious security problems in terms of authentication because it is impossible to keep track of the pseudonyms utilized by a user. To this purpose, the use of a *trusted authority* has been proposed.

Also in case of heterogeneous scenarios where mobile devices are available, some techniques similar to those used for vehicular communications can be considered [4, 6].

An approach that exploits the characteristics of nodes mobility could be inspired by CARAVAN [15]. Actually, CARAVAN applies to the case when devices move in *groups*.[3] These groups are created for a long period of time by devices located at within a short distance from each other. In this case, it is possible to select one of the nodes, called *group leader*, which is the responsible for the location update of the entire group. Accordingly: the number of messages exchanged for location updates decreases and thus, also the opportunity of eavesdropping decreases; only information about the location of groups is disclosed, and the specific location of an individual node is not revealed. In CARAVAN the problem of route un-traceability is not addressed since information flows through the wired communication infrastructure for most of the end-to-end path and thus, route backtracking is much more complex than in pure ad-hoc networks.

Regarding the approach of the second type, i.e. using routing solutions which do not imply disclosure of topology information, observe that flooding and geographical forwarding do not require exchange of location information between users. In case of geographical forwarding, however, there is still a problem, in that the sender needs to know the position of the destination. This can be performed, for example, by making a message be sent by the source S and flooded in the network to ask for the location of the destination (this message may be encrypted using the public key of the destination, D). Upon receiving such a message, the destination D will send the information about its own current position. This information will be encrypted using the public key of the source S. Observe that the use of flooding involves the flow of a large number of packets in the network. This causes an increase in privacy given that for an eavesdropper it will be almost impossible to distinguish the reply sent by the destination D from the other packets in the network. However, on the other hand, this leads to significant inefficiency in terms of both energy and bandwidth utilization.

In the above scenarios, only information about the location of mobile nodes is disclosed to allow update of their location and communication between them. In other words, only location privacy of individuals carrying mobile communication devices is menaced. When we consider scenarios involving wireless sensor networks, things are significantly different. In fact, sensors detect the presence of individuals (or assets) in a certain area regardless of the communication devices being carried. Messages related to the detection of the presence of a certain individual in a given area are generated by the relevant sensor nodes and forwarded to the sink(s) according to the wireless multihop communication paradigm. Even if the messages are encrypted, the presence of the above messages and their path in the network can be used by an attacker to violate the location privacy of the above individual.

An example of how the location privacy problem has been dealt with in wireless sensor networks, which are one of the main components of an IoT

[3] Observe that there are other scenarios where group mobility occurs [7].

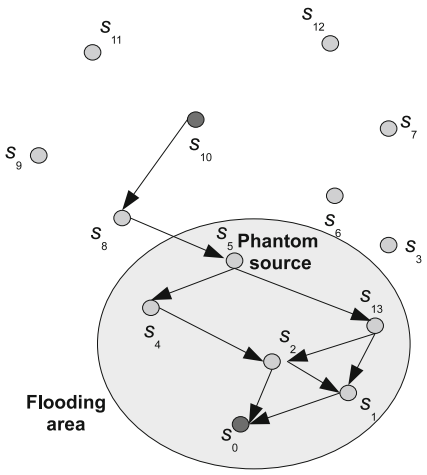


Fig. 5.4 Phantom routing

network, is represented by the *panda-hunter game* reference scenario [12]. Suppose that the *Save the Panda Organization* has deployed a wireless sensor network in a certain area to study and monitor the habits of pandas. Sensors detecting pandas in their proximity will generate messages that are forwarded to the sink. Therefore, the positions of sources are approximately equal to the position of pandas in that area.

In this scenario, a hunter carrying some equipment that enables him to detect the presence of a signal transmitted by sensor nodes can locate the pandas even if messages generated and transmitted by the nodes are encrypted and non readable by the hunter. Indeed, the hunter can go near the sink and then backtrace the messages generated by one of the sources until it reaches the source of such message flow. This example show how location privacy and routing are closely related in wireless sensor networks and thus, routing protocols should be privacy-aware. Location privacy can be improved by simply introducing a certain level of randomness in the routing algorithm. However, such randomness reduces the performance of routing in terms of both energy efficiency and delay. Thus, appropriate tradeoffs are needed [2].

One may think that another simple solution could be applying flooding; however, it has been demonstrated that flooding offers the least amount of location privacy protection [12].

An interesting solution is *phantom routing* as proposed in [12]. The source of a packet randomly selects another node in the network, called *phantom source*, and sends it the packet according to any routing scheme.[4] The phantom source will let the packet be flooded in the network. Actually, flooding is partial, being restricted to a portion of the network area, called *flooding area*, to increase efficiency. As an example, in Fig. 5.4 the source s_{10} selects node s_5 as phantom source and sends it

[4] A different phantom source is selected at each packet generation.

the packet which will be forwarded by node s_5 in the flooding area which contains nodes s_0, s_1, s_2, s_4, and s_{13}.

Even if flooding is partial, phantom routing still has efficiency problems. To overcome them, the Greedy Random Walk (GROW) protocol has been proposed in [21]. The basic idea of GROW is that at the i-th hop, the current relay forwards the packet towards the best next relay with probability p_i. Instead, any other neighbor node will be selected with probability $(1 - p_i)$. The value of p_i increases as i increases, so that the number of hops between the source and the destination is kept reasonably low. Accordingly, as compared to phantom routing, GROW achieves higher efficiency at the expenses of higher delay.

In RFID systems two types of problems can be encountered for what concerns privacy of things location. On the one hand, RFID tags are passive and only answer to queries without any control by the RFID tag owner. On the other hand, due to the broadcast nature of the wireless medium, eavesdroppers can listen to the tag replies and, as a consequence, infer the presence of a certain thing. Solutions to these problems are usually based on use of authorized readers so that RFID tags only answer to authenticated readers. In any case this approach, does not solve problems of eavesdropping. Also, use of authentication implies increase in complexity and costs and seems not to be suitable to RFID scenarios. Alternative solutions [11] use privacy negotiation schemes. Other schemes [17] create collisions on purpose in the wireless channel with the replies transmitted by the RFID tags so that when unauthenticated readers interrogate the tag, answers cannot be correctly received.

Encryption schemes could preserve RFID information but still allow malicious readers to detect the presence of the RFID tags scanned by the authorized reader. Accordingly a new set of solutions that use pseudo-noise as the signal transmitted by the reader can decrease the chance of intercepting the ongoing communication between the tag and the reader. Such noisy signal is modulated by the RFID tags and therefore, its transmission cannot be detected by malicious readers [16].

In order to ensure that the personal data collected is used only to support authorized services by authorized providers, solutions have been proposed that usually rely on a system called privacy broker [10]. The proxy interacts with the user on the one side and with the services on the other. Accordingly, it guarantees that the provider obtains only the information about the user which is strictly needed. The user can set the preferences of the proxy. When sensor networks and RFID systems are included in the network, then the proxy operates between them and the services. However, note that in this case, the individual cannot set and control the policies utilized by the privacy brokers. Moreover, observe that such solutions based on privacy proxies suffer from scalability problems.

5.4 Conclusions

Finding effective and credible solutions to privacy concerns is crucial to the actual success and diffusion of IoT services and applications. Indeed, people has proved

to be extremely cautious with respect to the introduction of new technologies when their introduction have menaced serious impact on their privacy. For example, the "Boycott Benetton" movement has forced Benetton to cancel the plan of tagging an entire new line of clothes with RFIDs.

In this paper, we have provided a survey of context privacy in heterogeneous IoT scenarios where ad hoc, RFID and sensor networks coexist. More specifically, we have described the problems related to application, correspondent, and location privacy and discussed some solutions proposed so far. Our survey of the current literature on the subject shows that, even if there has been a large research effort devoted to context privacy in the recent past, several issues are still open and require further investigation.

References

1. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A survey on wireless multimedia sensor networks. Comput. Netw. **51**(4), 921–960 (2007)
2. Armenia, S., Morabito, G., Palazzo, S.: Analysis of location privacy/energy efficiency tradeoffs in wireless sensor networks. In: Proceedings of IFIP Networking, Atlanta, GA, USA, May (2007)
3. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
4. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)
5. Choi, H., McDanial, P., Porta, T.F.L.: Privacy preserving communication in manets. In: Proceedings of 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and networks (SECON), San Diego, California, USA, June (2007)
6. Freudiger, J., Raya, M., Flegyhzi, M., Papadimitratos, P., Hubaux, J.-P.: Mix-zones for location privacy in vehicular networks. In: Proceedings of First International Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS), Vancouver, Canada, Aug (2007)
7. Galluccio, L., Morabito, G., Palazzo, S.: Spontaneous group management in mobile ad-hoc networks. Wirel. Netw. **10**(4), 423–438 (2004)
8. Goldschlag, D., Reed, M., Syverson, P.: Onion routing. Commun. ACM **42**(2), 39–41 (1999)
9. Kong, J., Hong, X.: ANODR: Anonymous on demand routing with untraceable routes for mobile ad hoc networks. In: Proceedings ACM Mobihoc 2003, Annapolis, MD, USA, June (2003)
10. Lioudakis, G.V., Koutsoloukas, E.A., Dellas, N., Kapellaki, S., Prezerakos, G.N., Kaklamani, D.I., Venieris, I.S.: A proxy for privacy: the discreet box. EUROCON 2007. Warsaw, Poland, September (2007)
11. Medaglia, C.M., Serbanati, A.: An overview of privacy and security issues in the internet of things. In: Proceedings of TIWDC 2009, Pula, Italy, September (2009)
12. Ozturk, C., Zhang, Y., Trappe, W., Ott, M.: Source-location privacy for networks of energy-constrained sensors. In: Proceedings of IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems (WSTFEUS), Vienna, Austria, May (2004)
13. Parent, W.: Privacy, morality and the law. Philos. Public Aff.**12**, 269–288 (1983)
14. Reed, M.G., Member, Syverson, P.F., Goldschlag, D.M.: Anonymous connections and onion routing. IEEE JSAC **16**(4), 482–494 (1998)

15. Sampigethaya, K., Huang, L., Li, M., Poovendran, R., Matsuura, K., Sezaki, K.: CARAVAN: Providing location privacy in vanets. In: Proceedings of Embedded Security in Cars (ESCAR), Cologne, Germany, November (2005)
16. Savry, O., Pebay-Peyroula, F., Dehmas, F., Robert, G., Reverdy, J.: Rfid noisy reader: How to prevent from eavesdropping on the communication? In: Proceedings of Workshop on Cryptographic Hardware and Embedded Systems 2007. Vienna, Austria, September (2007)
17. Savry, O., Vacherand, F.: Security and privacy protection of contactless devices. In: Proceedings of TIWDC 2009. Pula, Italy, September (2009)
18. Shelby, O.:Embedded web services. IEEE Wireless Communications Magazine (to appear)
19. Wang, X.Y., Chen, S., Jajodia, S.: Tracking anonymous peer-to-peer voip calls on the internet. In: Proceedings of ACM Conference on Computer Communications Security (CCS 2005), Alexandria, Virginia, USA, November (2005)
20. Weiser, M.: The computer for the twenty-first century. Scie. Am. **265**(3), 94–104 (1991)
21. Xi, Y., Schwiebert, L., Shi, W.: Preserving source location privacy in monitoring-based wireless sensor networks. In: Proceedings of 20th International Conference on Parallel and Distributed Processing Symposium (IPDPS 2006), Rhodes, Greece, April (2006)

Chapter 6
Physical Layer Cryptography and Cognitive Networks

Lorenzo Mucchi, Luca Simone Ronga and Enrico Del Re

Abstract Recently the huge development of different and heterogeneous wireless communication systems raises the problem of growing spectrum scarcity. Cognitive radio tends to solve this problem by dynamically utilizing the spectrum. Security in cognitive radio network becomes a challenging issue, since more chances are given to attackers by cognitive radio technology compared to conventional wireless network. These weaknesses are introduced by the nature itself of cognitive radio, and they may cause serious impact to the network quality of service. However, to the authors' knowledge, there are no specific secure protocols for cognitive radio networks. This paper will discuss the vulnerabilities inherent to cognitive radio systems, identify novel types of abuse, classify attacks, and analyze their impact. Security solutions to mitigate such threats will be proposed and discussed. In particular, physical payer security will be taken into account. A new modulation technique, able to encrypt the radio signal without any a priori common secret between the two nodes, was previously proposed by the authors (Mucchi et al. Wireless Personal Communications (WPC) International Journal 51:67–80, 2008; Mucchi et al. Wireless Personal Commun. J. 2010). The information is modulated, at physical layer, by the thermal noise experienced by the link between two terminals. A loop scheme is designed for unique recovering of mutual information. This contribution improves the previous works by proposing the noise loop modulation as physical layer security technique for cognitive radio networks.

L. Mucchi (✉) · E. Del Re
Department of Electronics and Telecommunications, University of Florence,
Via Santa Marta 3, 50139 Florence, Italy
e-mail: lorenzo.mucchi@unifi.it

E. Del Re
e-mail: enrico.delre@unifi.it

L. S. Ronga
e-mail: luca.ronga@cnit.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_6,

Keywords Cognitive radio · Mutual information · Cryptography · Physical layer

6.1 Introduction

Cognitive radio (CR) technology presents a promising solution for the spectrum shortage in wireless networks [1, 2]. It enables the efficient utilization of limited available spectrum by defining two kinds of user in wireless networks: licensed user and unlicensed user [3]. In cognitive radio networks, the unlicensed user can use the spectrum which is not temporarily used by licensed users. When the licensed user appears to use the spectrum, unlicensed user should return it back and search for other spectrum to use. Since in cognitive radio networks the spectrum is being used dynamically, general schemes cannot satisfy the special network requirements. Specific protocols need to be designed in order to manage the dynamic frequency spectrum and to ensure the quality of service (QoS).

Compared to wired network, the nature of wireless network makes the security vulnerability unavoidable. In wireless network, signal has to be transmitted through an open media without real connection. The data might be eavesdropped and altered without notice and the channel might be jammed or overused by an adversary. These will obviously disturb the normal communication and impact the quality of service. Due to their intrinsic nature, cognitive radio technology introduces more chances to attackers. For example, spectrum sensing is a key characteristic of cognitive radio networks; its main scope is to scan a certain range of the spectrum to detect unoccupied spectrum [4–6]. Through this process, unlicensed (secondary) user can determine if the radio spectrum can be used or not. However, if the spectrum sensing result is modified maliciously, normal network activities will be disabled, up to cause the whole network traffic to break down.

A conventional wireless device can access only one area of the radio spectrum, while an intelligent cognitive radio device (CRD) can sense and identify white spaces or vacant areas in the spectrum that can be used for communications. CR enables smart reconfigurable transceivers to make optimal use of spectrum by seeking out uncrowded bands and tuning into them with software that can adapt to a wide range of different frequencies and modulation schemes. In other words, CRDs have been proposed as a way to reuse under-utilized spectrum and their flexibility is seen as a possible solution to spectrum scarcity. This allows unlicensed devices to be secondary users of the spectrum and use the frequency bands only if the licensed or primary user of the spectrum is not active in the band. Unfortunately, these are also new unique opportunities for malicious attackers; cognitive radio networks (CRNs) introduce an entire new class of threats which are not easy to mitigate. The physical and link layers of CRNs are very different from those of conventional wireless networks. The particular attributes of CRNs, such as cooperative spectrum sensing, incumbent- and self-coexistence mechanisms, ask for new security implications. Nevertheless, this topic has received far less attention than other areas of CR.

6.2 Security Requirements

Although security requirements may vary in different application environment, usually there are some common requirements providing basic safety controls:

Privacy keeping information confidential; preventing disclosure to unauthorized users;

Access control permitting only authorized users to access specified information and services;

Integrity providing assurance that information has not been tampered with during handling;

Authentication providing proof of the credentials of the originator of information or a participant in a service; identity may not be the only attribute to be authenticated: other attributes such as location, functionality, or capability may be more significant in certain circumstances; the identity is verified normally because he/she has an object (e.g., key or smart card or), knows a secret (e.g., password) or owns a personal biologic characteristic (e.g., fingerprint);

Authorization specify the actions that each user can do;

Non-repudiation preventing a participant in a service or transaction from denying having taken some specific action.

In the very next future the wireless communications will experience a deep change. General ubiquity, new context-aware applications and services, new network and terminal technologies, flexible spectrum management and dynamic reconfiguration will be some of the key features of this change. New technologies, facilities and capabilities will generate new requirements for security, even because users are more and more aware of the impact of these developments on their personal privacy.

While many security techniques developed in wired networks can be applied, the special characteristics of wireless networks call for innovative wireless security design. Since physical-layer security techniques can address directly such special wireless characteristics, they are helpful to provide boundary control, to enhance efficiency, as well as to assist upper-layer security techniques for innovative cross-layer security designs.

6.3 Wireless is Different from Wired and Cognitivity Makes It Even More Different

The term Cognitive Radio was first used by Mitola in [7]. Some of the key features that are typically associated with CR include [8]:

- *Maintains awareness* of surrounding environment and internal state
- *Adapts* to its environment to meet requirements and goals

- *Learns* from previous experiences to recognize conditions and enable faster reaction times
- *Anticipates* events in support of future decisions
- *Collaborates* with other devices to make decisions based on collective observations and knowledge.

Each of these characteristics could provide a new class of attacks for a wireless network [9, 10]. For example,

- *Maintains awareness* of surrounding environment and internal state $\Longrightarrow$ Opportunity for spoofing
- *Adapts* to its environment to meet requirements and goals $\Longrightarrow$ Opportunity to force desired changes in behavior in victim
- *Learns* from previous experiences to recognize conditions and enable faster reaction times $\Longrightarrow$ Opportunity to affect long-lasting impact on CR behavior
- *Anticipates* events in support of future decisions $\Longrightarrow$ Opportunity for long-lasting impact
- *Collaborates* with other devices to make decisions based on collective observations and knowledge $\Longrightarrow$ Opportunity to propagate attack through network.

The typology of attacks in cognitive radio network can be summarized into two big categories [11]:

Theft of privileges: This attack happens when the attacker wants to have access to spectrum with higher priority. It can be achieved by misleading other unlicensed users to believe there is a licensed user active in the spectrum area. As a result, the adversary can occupy the spectrum resource as long as he/she wants.

Denial of service: This attack happens when the adversary inhibits other unlicensed users from using the spectrum and causes the denial of service (DoS). As a serious result, malicious attack will extremely decrease the available bandwidth and break down the whole traffic. For example, in multi-channel environment, high traffic load may cause frequent exchange of control packets. If the adversary can saturate the control channel successfully, it can hinder the channel negotiation and allocation process.

Security in a cognitive network does not only mean providing that no intruders can access the network (outside attack, both passive and active) but also that an hypothetical intruder is properly and quickly detected (inside attack: intrusion, misuse and anomaly detection).

The security features of a cognitive radio network should be based on both protection and detection, and each protocol layer should be considered due to the specific characteristic.

6.4 Layered Security

While the advantages of wireless cognitive networks are tremendous, the security issues are real. Without physical security that can be used to protect wired

networks, wireless users need to protect their networks with other tools that can provide the same level of security as wired networks. These solutions can be layered to provide the level of security required for any user or organization. Like all IT-based security, cognitive WLAN security should be handled in layers. This provides several advantages: stronger overall security, the ability to block access at multiple layers of the network, and flexibility in selecting the cost/benefit ratio of the desired solution. The layered security approach also provides the benefit of selecting the desired level of security, compared against the costs of adding additional layers.

Physical layer security (PHY-sec) is built into wireless equipment, and is essentially free (except for the cost of configuring and maintaining encryption keys) and may be adequate for a home user who wants to keep out the casual intruder. PHY-sec is extremely important for cognitive radio networks because it is the first access to the network, but it must be designed in order to be applied to different physical layers as can be in a cognitive network. This means that the PHY-sec should take into account techniques which are flexible enough to be applied to different standard and physical layers without relying to upper layers.

6.5 Physical Layer Security

Security at physical layer is nowadays mainly intended as the use of a spread spectrum techniques (frequency hopping, direct sequence coding, etc.) in order to avoid the eavesdropping. Eavesdropping at the physical layer refers to hiding the very existence of a node or the fact that communication was even taking place from an adversary. This means that the communication of the legal user is already on, i.e., that the authentication of the legal user has been already performed. Moreover, scrambling the information data with a code does not assure a totally secure channel, but just a long-time activity before getting the code by an unwanted listener, i.e., the security is moved on the quantity of resources (hardware, time, etc.) that the unwanted listener must have in order to get the information.

It is well known that classical encryption techniques have only unproven complexity-based secrecy [12]. We also know that strong information-theoretic secrecy or perfect secrecy is achievable by quantum cryptography based on some special quantum effects such as intrusion detection and impossibility of signal clone [13]. Unfortunately, the dependence on such effects results in extremely low transmission efficiency because weak signals have to be used. One of the recent attempts on specifying secret channel capacity is [14], where the MIMO secret channel capacity is analyzed under the assumption that the adversary does not know even his own channel. Unfortunately, such an assumption does not seem practical if considering deconvolution or blind deconvolution techniques. Moreover, such techniques are not low-complex, due to the fact that they need a high number of antennas on both sides of the radio link to correctly work. As a matter of

fact, almost all existing results on secret channel capacity are based on some kinds of assumptions that appear impractical [15–17]. It has been a challenge in information theory for decades to find practical ways to realize information-theoretic secrecy.

Moreover, one of the most weak point of wireless networks is the initial data exchange for authentication and access procedures. Initially some data must be exchanged in a non-secure radio channel or anyway by sharing a common known cryptographic key. At the moment, no physical layer techniques are present in the literature which can efficiently create a secure wireless channel for initial data exchanging between two network nodes/terminals without a priori common knowledge.

The main need is to exchange cryptographic keys between two users along an intrinsically secure radio channel. As stated before, classical encryption techniques have only unproven complexity-based secrecy. Information-theoretic secrecy or perfect secrecy is achievable by quantum cryptography, but unfortunately this technique is suitable (when applicable) only to optical networks. Derived by the Ultra WideBand (UWB) communications, the radio channel identifier (RCI) is another promising technique. But, again, the information exchanging process is not absolutely secure.

A review of the main physical layer techniques for secret key distribution is reported in the following sections. Then the novel technique is described and detailed.

6.5.1 Quantum Cryptography

The quantum cryptography [18], or quantum key distribution (QKD), method uses quantum mechanics to guarantee secure communication. It enables two parties to produce a shared random bit string known only to them, which can be used as a key to encrypt and decrypt messages. The process of measuring a quantum system in general disturbs the system and thus render the information unreadable. A third party trying to eavesdrop on the key must in some way measure it, thus introducing detectable anomalies. Nowadays, quantum cryptography is only used to produce and distribute a key, not to transmit any message data. This method is suitable only for optical networks. If the optical network is wireless, a high-SNR line of sight is mandatory, which makes the method not properly flexible for real applications.

6.5.2 Channel Identifier

This technique [19] is based on transmitting a short pulse and measuring the channel impulse response. The impulse response of the channel between the two

users can be the encryption key of the transmission. The procedure can be summarized as follows:

- Each radio terminal (the two legal users, for example) transmits an identical signal.
- Each user observes the channel impulse response of the channel.
- The users exchange some information about what they observed, e.g., part of the sampled channel impulse response that have been observed previously.
- The users use a public channel to determine the channel identifier (the encryption key, i.e., the shared secret).
- The users begin communicating data, encrypting it using the channel identifier as a key.

Mainly, this method is based on the assumption that a third radio in a different location will observe a different channel impulse response, and that the channel is reciprocal. If two users transmit the same pulse and possess identical receivers, then the observed channel impulse response can act as a source of common randomness for secret key generation. The main drawback of this method is that the two users still have to share some information through a non-secure channel, and again the users have to share a common knowledge in other to build a secure shared secret.

6.5.3 MIMO

One of the recent attempts on specifying secret channel capacity by using MIMO (Multiple Input Multiple Output) technique [20]. The mobile terminals are equipped with multiple antennas, N transmitting and M receiving. The symbol is encrypted by using the matrix $N \times M$ of channel impulse responses provided by the multiple channels [21, 22]. A valid way to guarantee a high error rate for the eavesdropper is to prevent it from channel estimation. In terms of channel estimation, the legal receiver has no advantage over the eavesdropper. Therefore, our objective is to design a transmission scheme so that the legal receiver can detect signals without channel knowledge, which can be realized by shifting the channel estimation task from the receiver to the transmitter. Once the transmitter has the channel knowledge, it can adjust the MIMO transmission so that the receiver does not need to estimate channel in order for symbol estimation. Reciprocity of the forward and backward channels is normally used. The receiver first transmits a pilot signal to the transmitter using the same carrier frequency as the secret channel, during which the transmitter can estimate the backward channel, and use it for array transmission. Such techniques are not low-complex, due to the fact that they need a high number of antennas on both sides of the radio link to correctly and securely work. Moreover, the two legal hosts are forced to exchange information (the initial pilot channel, in a non-secure channel) which can be exploited by the eavesdropper.

6.5.4 The Noise Loop Modulation: How It Works

Finally, it is important to highlight that the proposed technique does not assure any mechanism of identification of the user. The identification process must be controlled by the higher level, but nothing else than this because the information is made secure by the physical layer itself, i.e., the transmission cannot be demodulated by an unwanted user.

The information is modulated, at physical layer, by the thermal noise experienced by the link between two terminals. A loop scheme is designed for unique recovering of mutual information. All results show that the mutual information exchanged by the two legal terminals cannot be demodulated or denied by a third terminal. At the same time the two legal users do not suffer from the presence or not of a third unwanted user from the performance point of view.

6.6 The Noise-Loop Modulation

A novel idea for low-complex intrinsic secure radio link without any a priori knowledge was previously proposed by the authors [9, 10]. One of the main advantages of this new technique is the possibility to have a physically secure radio channel for wireless systems without any a priori common knowledge between legal source and destination. This feature is particularly useful for future wireless cognitive networks. Due to the intrinsic unique nature of the thermal noise along each radio link between the specific transmitter and the specific receiver, this novel technique is particularly suitable for secure communications and privacy in cognitive networks because it can easily applied to each kind of physical layers.

In order to remind the proposed technique [9], let us suppose two terminals exchanging information: terminal 1 and terminal 2. Two different channels are considered: one for the link from terminal 1 to terminal 2 and one for the reverse link. The two channels are considered on different frequency bands and the thermal noise on one link is considered uncorrelated with the other. Each link is modeled as a conventional AWGN channel.

6.6.1 Symbols in the Paper

The following symbols have been adopted in the paper:

b_i binary antipodal ($b_i \in \{-1;\ +1\}$) information signal originating form terminal i,

$n_i(t)$ Gaussian, white, continuous time random processes modeling the received noise at the receiver on terminal i, characterized by zero mean and variance σ_n^2,

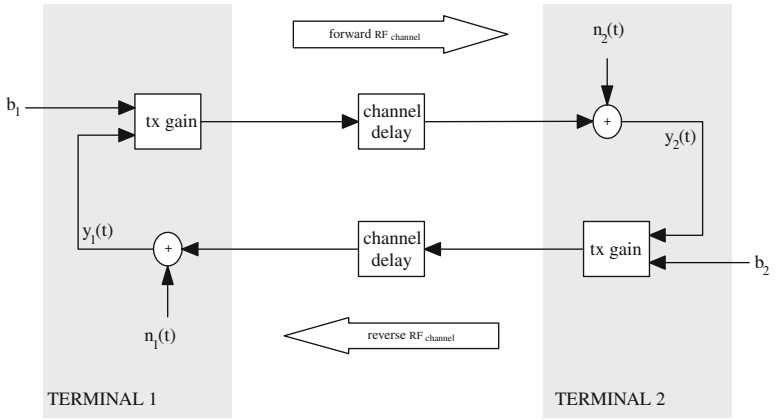


Fig. 6.1 Noise-loop chain scheme. Two terminals communicates by using the noise loop. The parameters in the scheme are explained at the beginning of Sect. 6.1

α_i global link gain $(0<\alpha<1)$ for the signal generated from terminal i. It includes transmission gain, path loss and channel response. It is also supposed to be known by the receivers,

τ_p propagation delay for the channel. It is assumed without loss of generality that both forward (1 to 2) and reverse (2 to 1) links have the same delay,

$y_i(t)$ baseband received signal available at the terminal i.

6.6.2 System Model

In this simple transmission system the terminal operations are described in Fig. 6.1. The signal from the inbound channel is modulated by the information and re-transmitted on the outbound channel. The reception is obtained by extracting the sign of the $2\tau_p$-delayed autocorrelation term of the incoming signal, multiplied by the own informative bit. The whole process is detailed in the following sections.

Let us focalize without loss of generality on the user terminal 1.

The reception, i.e., the extraction of the information bit b_2, is obtained by extracting the sign of the $2\tau_p$-delayed autocorrelation term of the incoming signal, multiplied by the own informative bit b_1.

Due to the additive nature of the model, the received signals $y_1(t)$ available at terminal 1, after infinite loop iterations, is defined by the following series:

$$y_1(t) = \sum_{j=0}^{\infty}(b_1 b_2 \alpha_1 \alpha_2)^j n_1(t - 2j\tau_p) + \sum_{j=0}^{\infty}(b_1 b_2 \alpha_1 \alpha_2)^j b_2 \alpha_2 n_2(t - (2j+1)\tau_p) \tag{6.1}$$

An analogue expression can be obtained for $y_2(t)$ simply exchanging the subscript 1 and 2 in (6.1).

If the noise processes $n_i(t)$ are white on a unlimited bandwidth, then:

$$E\left[n_i(t)n_j(t-\tau)\right] = \begin{cases} \delta(\tau)\sigma_n^2 & i=j \\ 0 & i \neq j \end{cases} \tag{6.2}$$

The structure of the signal in (6.1) draw our attention in the shifted correlation term

$$y_1(t-2\tau_p)y_1(t) \tag{6.3}$$

By resorting the terms obtained by the expansion of the above expression, the expectation of the autocorrelation in (6.3) can be written as following:

$$\begin{aligned} E[y_1(t-2\tau_p)y_1(t)] = b_1 b_2 \sigma_n^2 (1+\alpha_2^2) \sum_{j=0}^{\infty} (\alpha_1\alpha_2)^{2j+1} \\ + E[\text{residual cross correlation terms}] \end{aligned} \tag{6.4}$$

The last term in (6.4) is null for an ideal AWGN channel, so the described autocorrelation term is dominated by the information bearing $b_1 b_2$ term, weighted by a term which is constant in the stationary case. The term contains the information of both terminals. Since terminal 1 is depicted to detect information bits of terminal 2, it is sufficient to perform a post-multiplication by b_1 in order to estimate the sign of b_2.

6.7 Performance Analysis

The performance of the proposed receiver in terms of bit error probability is related to the first and second order statistics of (6.3). The distribution of the unpredictable noise process, however, is no longer Gaussian [9].

The mean value of the decision variable (6.3) is

$$E[y_1(t)y_1(t-2\tau_p)] = b_1 b_2 \alpha_1 \alpha_2 \frac{\sigma_n^2(1+\alpha_2^2)}{1-\alpha_1^2\alpha_2^2} \tag{6.5}$$

where $\sigma_n^2 = \text{var}[n_1(t)] = \text{var}[n_2(t)]$ is the variance of the thermal noise processes involved in the loop. The relation between σ_n and σ is

$$\sigma^2 = \text{var}[y_1(t)] = \text{var}[y_1(t-2\tau_p)] = \frac{\sigma_n^2(1+\alpha_2^2)}{1-\alpha_1^2\alpha_2^2}$$

For the sake of simplicity, let us assume hereby that $\alpha_1 = \alpha_2 = \alpha$, assumed that $0 < \alpha < 1$.

Supposing a binary antipodal signalling (BPSK) modulation, the receiver 1 demodulates the bit b_2 by deciding on the sign of the decision variable $d = b_1 \cdot s$ where $s = E[z] = E[y_1(t)y_1(t - 2\tau_p)] = b_1 b_2 |\rho| \sigma^2$. Thus, an error occurs when $d > 0$ but $b_2 = -1$ and $d < 0$ but $b_2 = 1$.

Due to the symmetry of the pdfs, the total probability of error P_e for the receiver terminal 1 can be written as

$$P_e = P_e(\alpha) = 2 \int_{\alpha^2}^{\infty} \frac{e^{\frac{-z'}{1-\alpha^4}}}{\pi\sqrt{1-\alpha^4}} K_0\left(\left|\frac{z'}{1-\alpha^4}\right|\right) dz' \tag{6.6}$$

As a first important result, it can be highlighted that the probability of error does not depend on the noise variance but only on the loop gain α.

6.8 Probability of Detection of a Third Unwanted Listener

Let us assume hereby that a third unwanted user is listening the transmission of terminal 1. The terminal 3 can be supposed, without loss of generality, to have a different propagation delay $\tau_{p3} \neq \tau_p$ and an independent thermal noise process $n_3(t) \neq \{n_1(t), n_2(t)\}$.

The signal received by the unwanted listener can be written as

$$\begin{aligned} \bar{y}_1(t) = y_1(t) + n_3(t) &= n_3(t) + n_1(t) b_2 \alpha_2 n_2(t - \tau_p) \\ &+ b_1 b_2 \alpha_1 \alpha_2 n_1(t - 2\tau_p) + b_1 \alpha_1 b_2^2 \alpha_2^2 n_2(t - 3\tau_p) \\ &+ b_1^2 b_2^2 \alpha_1^2 \alpha_2^2 n_1(t - 4\tau_p) + b_1^2 b_2^3 \alpha_1^2 \alpha_2^3 n_2(t - 5\tau_p) + \cdots \end{aligned} \tag{6.7}$$

The mean value of the $2\tau_{p3}$-shifted correlation $\bar{y}_1(t)\bar{y}_1(t - 2\tau_{p3})$ can be derived to be

$$E[\bar{y}_1(t)\bar{y}_1(t - 2\tau_{p3})] = \begin{cases} \frac{b_1^k b_2^k \sigma_n^2 (\alpha_1 \alpha_2)^k (1+\alpha_2^2)}{1-\alpha_1^2 \alpha_2^2} & \text{if } \tau_{p3} = k\tau_p,\ k \in \mathcal{N} \\ 0 & \text{if } \tau_{p3} \neq k\tau_p,\ k \in \mathcal{N} \end{cases} \tag{6.8}$$

The decision variable of the third unwanted party depends on the multiplication of the information bits b_1 and b_2, both unknown at the unwanted listener.

This result means exactly that the third unwanted listener is not able at all to demodulate the information exchanged between users 1 and 2. This impossibility is intrinsic in the modulation method at physical layer level.

6.9 Probability of Denial of Service by a Third Unwanted User

The user n.3 tries to disturb as much as possible the radio link between legal users n.1 and n.2 by actively starting a noise loop modulated communication towards user n.1. The received signal of user n.1 is now

$$y_1(n) - K_{12}y_1(n-2k) - K_{13}y_1(n-2h) = n_1(n) + K_2 n_2(n-k) + K_3 n_3(n-h) \tag{6.9}$$

where $K_{ij} = K_i K_j$ with $K_i = b_i\alpha_i$, while k and h are the propagation delays of the radio links between users 1–2 and 1–3, respectively. In this case the noise term is $e(n) = n_1(n) + K_2 n_2(n-k) + K_3 n_3(n-h)$ and it is still a white Gaussian process $e(n) \sim N(0, \sigma_e^2)$ with

$$\sigma_e^2 = (1 + K_2^2 + K_3^2)\sigma_n^2 = (1 + \alpha_2^2 + \alpha_3^2)\sigma_n^2 \tag{6.10}$$

Equation 6.9 can be written as a stochastic autoregressive process

$$y_1(n) - K_{12}y_1(n-2k) - K_{13}y_1(n-2h) = e(n) \tag{6.11}$$

that can be solved by using the Yule–Walker equations [23]

$$\begin{cases} R_{y_1y_1}(2k) - K_{13}R_{y_1y_1}(2(h-k)) - K_{12}R_{y_1y_1}(0) = 0 \\ R_{y_1y_1}(2h) - K_{12}R_{y_1y_1}(2(h-k)) - K_{13}R_{y_1y_1}(0) = 0 \\ -K_{13}R_{y_1y_1}(2k) - K_{12}R_{y_1y_1}(2h) + R_{y_1y_1}(2(h-k)) = 0 \\ -K_{12}R_{y_1y_1}(2k) - K_{13}R_{y_1y_1}(2h) + R_{y_1y_1}(0) = \sigma_e^2 \end{cases} \tag{6.12}$$

We aim to extract the decision variable of user n.1 $R_{y_1y_1}(2k)$ in order to see if the legal user n.1 suffers from the presence of the unwanted user n.3 communication. By solving the system we have

$$R_{y_1y_1}(2k) = \begin{cases} b_1 b_2 \frac{\alpha_1\alpha_2(1+\alpha_2^2+\alpha_3^2)(1+\alpha_1^2\alpha_3^2-\alpha_2^2\alpha_3^2)}{(1-\alpha_1^2\alpha_2^2-\alpha_1^2\alpha_3^2)^2}\sigma_n^2 & \text{if } h \neq k,\ h \neq 2k,\ h \neq k/2 \\ b_1\alpha_1(b_2\alpha_2 + b_3\alpha_3)\frac{\sigma_e^2}{1-\alpha_1^2\alpha_2^2-\alpha_1^2\alpha_3^2} & \text{if } h = k \\ b_1 b_2 \alpha_1\alpha_2 \frac{\sigma_e^2}{(1-b_1b_3\alpha_1\alpha_3)(1-\alpha_1^2\alpha_2^2-\alpha_1^2\alpha_3^2)} & \text{if } h = 2k \\ 0 & \text{if } h = k/2 \end{cases} \tag{6.13}$$

The sign of the first term of Eq. 6.13 depends on the sign of b_1b_2 and not on b_3, so the legal user n.1 can correctly demodulate the information coming from user n.2 without any disturb from user n.3. The second term of Eq. 6.13 $R_{y_1y_1}(2k) = b_1\alpha_1(b_2\alpha_2 + b_3\alpha_3)\frac{\sigma_e^2}{1-\alpha_1^2\alpha_2^2-\alpha_1^2\alpha_3^2}$ depends on the information bit b_3 so invalidating the radio link 1–2, but it is valid only when the propagation

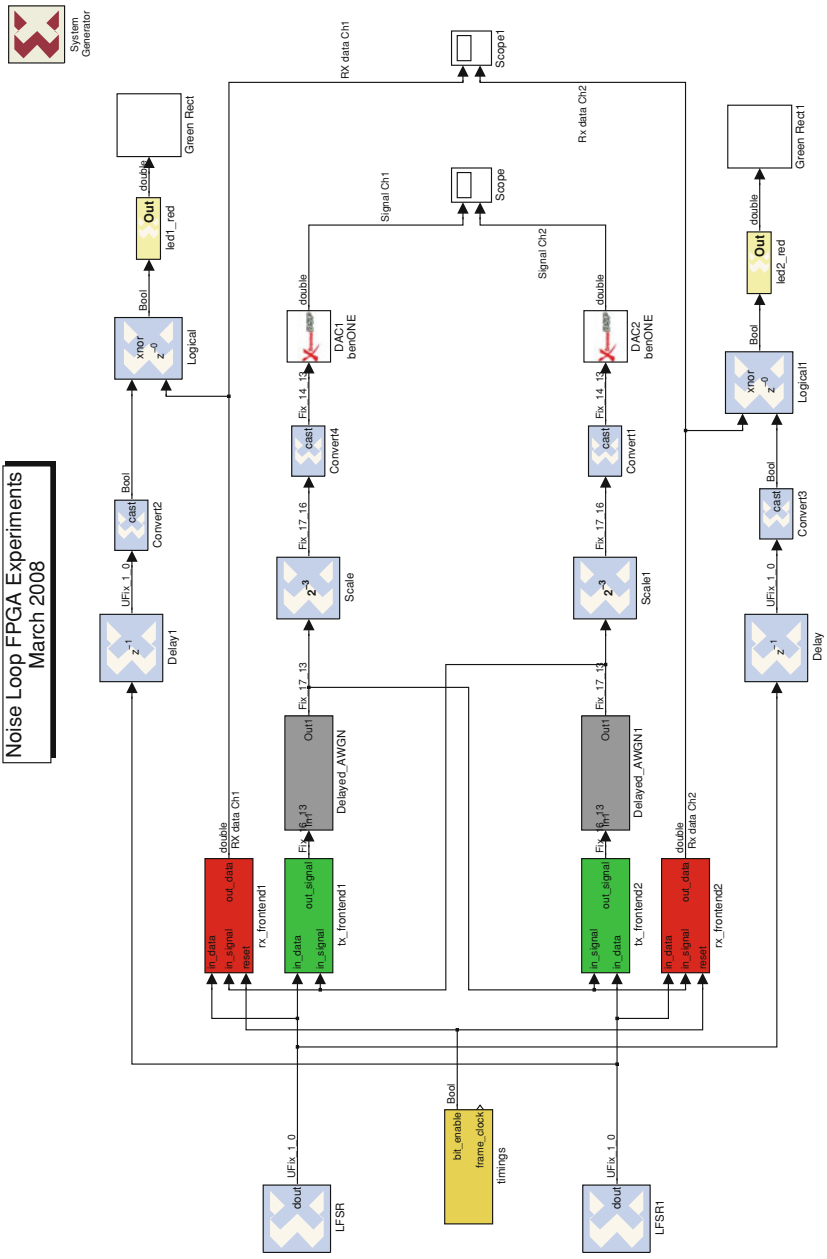


Fig. 6.2 FPGA implementation: Simulink model of noise-loop transmission chain

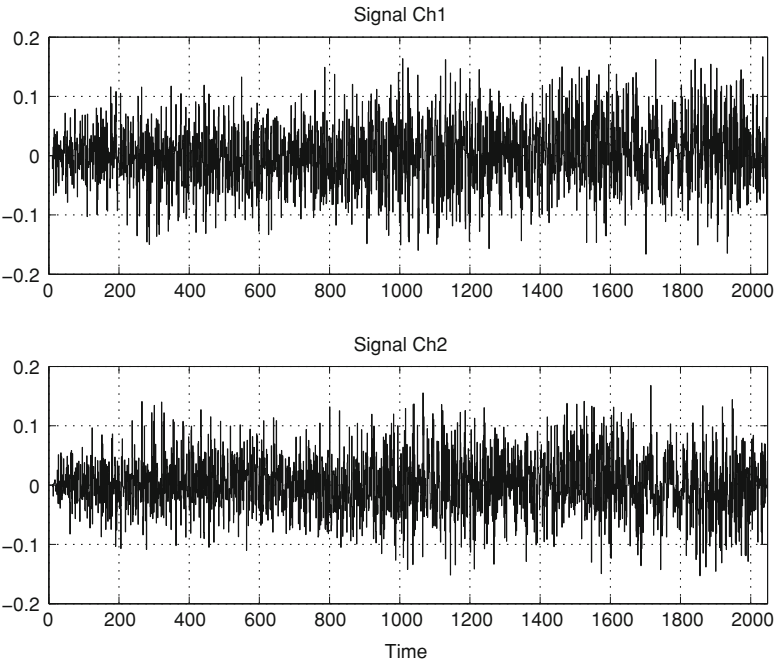


Fig. 6.3 FPGA implementation: baseband signals for forward and reverse channels

delay of the radio link between users 1–2 and 1–3 is exactly the same. This condition is almost impossible in real wireless scenarios. The third term $R_{y_1y_1}(2k) = b_1b_2\alpha_1\alpha_2\frac{\sigma_e^2}{(1-b_1b_3\alpha_1\alpha_3)(1-\alpha_1^2\alpha_2^2-\alpha_1^2\alpha_3^2)}$ does not imply any denial of service, while the fourth term does because the improbable condition $h = k/2$ causes the decision variable to be zero. The presence of two very particular points which can cause denial of service (DoS) should not create panic because those situations are incredibly improbable and moreover the two legal users, once the noise loop modulation is started, can exchange a locally generated additional delay in order to avoid such dangerous situations.

6.10 FPGA Implementation

The proposed TX/RX scheme has been implemented on a Xilinx Virtex II FPGA to evaluate the real computational complexity and to prove the validity of data detection on a fixed point signal processing system. The transmission, reception and baseband processing branches for two terminals have been implemented in VHDL using Xilinx System Generator tool. The Matworks Simulink model of the

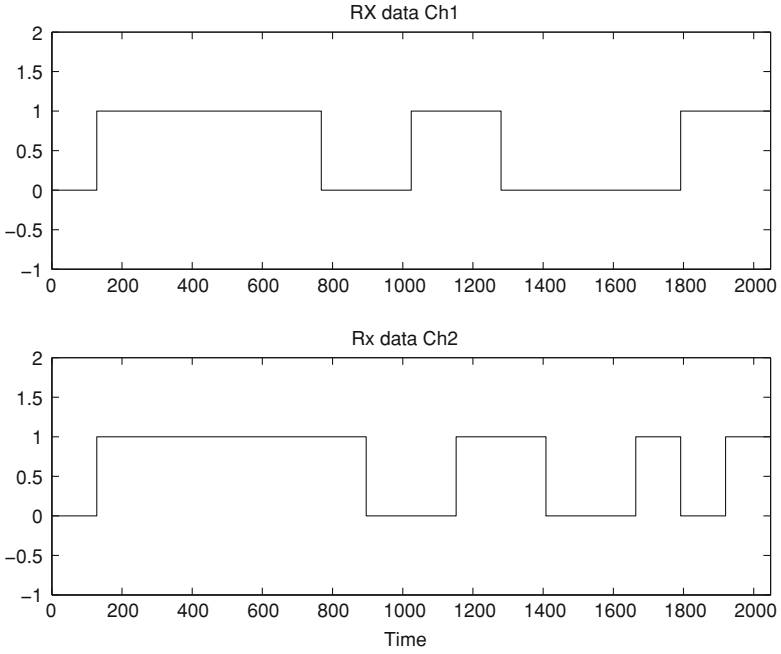


Fig. 6.4 FPGA implementation: detected data from both receivers

Table 6.1 FPGA utilization (total and fraction of Xilinx Virtex II xc2vp30-5ff1152 resources)

FPGA resource	TX module	RX module
Multipliers (MULT)	1 (0.7%)	3 (2.2%)
Look-up tables (FGs)	136 (0.4%)	42 (0.1%)
Arithmetic logic (CYs)	75 (0.2%)	40 (0.1%)
Storage elements (DFFs)	68 (0.2%)	165 (0.6%)

implemented chain is represented in Fig. 6.2. The model includes two PN-sequence generators to emulate the transmitted data from both terminals, a timing section, the TX and RX frontends, a delayed AWGN emulator, and other performance evaluating blocks. The baseband signal generated by the two noise-loop terminals are represented in Fig. 6.3 for both forward and reverse channels (respectively Ch1 and Ch2). The detected data are reported in Fig. 6.4. No visible correlation can be found with the baseband signal of Fig. 6.3, confirming the assumptions made in the theoretical sections of this work.

The complexity of the proposed implementation is reported in Table 6.1. As shown both the transmitter and receiver blocks use a very small portion of the used FPGA, though framing and synchronization have not been addressed yet.

6.11 Conclusions

This paper presents a survey on physical layer security in cognitive radio networks. Challenging problems are outlined due to the characteristic of accessing spectrum dynamically in cognitive radio networks. First, general security problems are analyzed in wireless networks because wireless media is the common characteristic in cognitive radio networks. Then, security problems, especially relative to cognitive radio network, are discussed.

A possible (new) technique for physical layer security in cognitive radio networks, previously proposed in [9, 10], is here discussed. Potential applications of the proposed techniques are found in wireless communications systems where an initial shared secret is not available.

References

1. Staple, G., Werbach, K.: The end of spectrum scarcity. IEEE Spectr. **41**(3), 48–52 (2004)
2. Haykin, S.: Cognitive radio: brain-empowered wireless communications. IEEE J. Selected Areas in Commun. **23**(2), 201–220 (2005)
3. Akyildiz, I.F., Lee, W.-Y., Vuran, M.C., Mohantly, S.: Next generation/dynamic spectrum access/cognitive radio wireless network: a survey. Comput. Networks **50**, 2127–2159 (2006)
4. Burbank, J.L.: Security in cognitive radio networks: the required evolution in approaches to wireless network security. In: 3rd International Conference on CrownCom 2008, pp. 1–7, 15–17 May 2008
5. Akyildiz, I.F., Lee, W., Vuran, M.C., Mohanty, S.: A survey on spectrum management in cognitive radio networks. IEEE Commun. Mag. **46**, 40–80 (2008)
6. Kaligineedi, P., Khabbazian, M., Bhargava, V.K.: Secure cooperative sensing techniques for cognitive radio systems. In: IEEE International Conference on Communications. ICC '08 (2008)
7. Mitola III, J., MaGuire Jr., GQ.: Cognitive radio: making software radios more personal. IEEE Personal Commun. **6**(4), 13–18 (1999)
8. IEEE 802 Tutorial: Cognitive Radio, Scott Seidel, Raytheon, Presented at the IEEE 802 Plenary, 18 July 2005
9. Mucchi, L., Ronga, L.S., Cipriani, L.: A new modulation for intrinsically secure radio channel in wireless systems, published in the special issue "Information Security and Data Protection in Future Generation Communication and Networking" of the Wireless Personal Communications (WPC) International Journal, by Springer; published online in 2008, printed in vol. 51, pp. 67–80, Number 1/October, 2009. doi:10.1007/s11277-008-9609-8
10. Mucchi, L., Ronga, L.S., Del Re, E.: A novel approach for physical layer cryptography in wireless networks. Wireless Personal Commun. J. Springer (2010), to be published. doi: 10.1007/s11277-010-9950-6
11. Chen, R., Park, J., Hou, Y., Reed, J.H.: Toward secure distributed spectrum sensing in cognitive radio networks. IEEE Commun. Mag. **46**, 50–55 (2008)
12. Shannon, C.: Communication theory of secrecy systems. Bell Syst. Tech. J. **29**, 656–715 (1949)
13. Bennett, C.H., Brassard, G.: In: IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, pp. 175–179 (1984)
14. Hero III, A.O.: Secure space–time communication. IEEE Trans. Inform. Theory **49**(12), 3235–3249 (2003)

15. Maurer, U.: Secret key agreement by public discussion from common information. IEEE Trans. Inform. Theory **39**(3), 733–742 (1993)
16. Wyner, A.D.: The wire-tap channel. Bell Syst. Tech. J. **54**(8), 1355–1387 (1975)
17. Csiszar, I., Korner, J.: Broadcast channels with confidential messages. IEEE Trans. Inform. Theory **24**(3), 339–348 (1978)
18. Sharbaf, M.S.: Quantum cryptography: a new generation of information technology security system, information technology: new generations. In: 2009. Sixth International Conference on ITNG '09, pp. 1644–1648, 27–29 April 2009
19. Wilson, R., Tse, D., Scholtz, R.A.: Channel identification: secret sharing using reciprocity in ultrawideband channels. IEEE Trans. Inform. Forencis Security 2(3), 364–375 (2007)
20. Hyungjin, K., Villasenor, J.D.: Secure MIMO communications in a system with equal numbers of transmit and receive antennas. Commun. Lett. IEEE **12**(5), 386–388 (2008)
21. Li, X., Ratazzi, EP.: MIMO transmissions with information-theoretic secrecy for secret-key agreement in wireless networks. In: IEEE Military Communications Conference (MILCOM'2005), Atlantic City, NJ, 17–20 October 2005
22. Mohammadi, M.S.: MIMO minimum leakage. Physically secure wireless data transmission. In: International Conference on Application of Information and Communication Technologies, 2009. AICT 2009, pp. 1–5, 14–16 October 2009
23. Pollock, D.S.G.: A Handbook of Time-Series Analysis, Signal Processing and Dynamics. Academic Press, New York, ISBN 0-12-560990-6 (1999)

Chapter 7
Gram-Schmidt Orthogonalization on Encrypted Vectors

Pierluigi Failla and Mauro Barni

Abstract In this paper we address a privacy preserving version of the well known Gram-Schmidt orthogonalization procedure. Specifically, we propose a building block for secure multiparty computation, that is able to orthogonalize a set of componentwise encrypted vectors. Our setting is the following: Bob needs to compute this orthogonalization on some vectors encrypted with the public key of Alice. Hence, our intent is not to propose a stand-alone protocol to solve a specific scenario or a specific application, but rather to develop a sub-protocol to be embedded in more complex algorithms or protocols where the vectors to be orthogonalized can be the result of previous computations. We show that our protocol is secure in the *honest but curious model* and evaluate its computation complexity.

Keywords Secure multi-party computation · Homomorphic cryptography

7.1 Introduction

The classical way to protect sensitive information from misuse is to encrypt it as soon as the information is generated and to store it in an encrypted form. However, when the information needs to be processed, it is necessary to decrypt it, hence, creating a weakness in the security of the whole process. The problem with the classical approach is the assumption that the owner of the data and the party in charge of processing it trust each other: the encryption layer is used only to protect

P. Failla (✉) · M. Barni
Department of Information Engineering, University of Siena, Via Roma 56, 53100 Siena, Italy
e-mail: pierluigi.failla@gmail.com

M. Barni
e-mail: barni@dii.unisi.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_7,

the data against third parties. In many cases, however, the owner of the information may not trust the third parties that are asked to manipulate the sensitive informations. In this scenario, the possibility of applying particular cryptographic techniques to process encrypted data has received a considerable attention in the last years. The problem of computing with encrypted data has been intensively studied in the past 30 years [21]. Following that direction, researchers developed many protocols to be applied in applications where the privacy and the security of the inputs are crucial. The proposed applications range from heuristic search in encrypted graphs [12]; ElectroCardioGram (ECG) classification [4]; data mining [1]; face recognition [11]; remote diagnosis [7].

In this paper, we consider a scenario in which two parties are interested in computing a given functionality in a privacy preserving way, but this functionality needs a sub-protocol that computes the Gram-Schmidt orthogonalization on encrypted vectors. Our intent is to study this particular sub-protocol, giving a detailed description comprehensive of security proof and complexity evaluation.

To the best of our knowledge, this problem has never been addressed so far. Therefore, this work focuses on the problem of developing a protocol that realizes the Gram-Schmidt procedure in a privacy preserving fashion. There are a lot of applications in which this kind of sub-protocol could be embedded as a basic privacy preserving primitive, including: QR decomposition [13]; linear least squares problems [6]; face recognition [24]; improving performances of neural networks [19]; wavelets computation [9]; principal component analysis [22] and image compression [18].

7.2 Signal Processing in the Encrypted Domain

The classical security model is targeted towards protecting the communication between two trusted parties against a third malicious party. In such cases it is sufficient to secure the transmission layer that stays on top of the processing blocks. Most of the applications today, work in this way, for instance when you log in a website, only the transmission is protected and on the other side the website is able to interpret your data in plain form while eventual thirds parties can see only an encrypted communication.

In the last years, the number of applications in which the classical model in not longer adequate has considerably increased since there are several non-trusted parties involved in the process of communication, distribution and processing data. Consider, for example, a remote diagnosis service (say Bob) where a non-trusted party is asked to process some medical data (owned by Alice) to provide a preliminary diagnosis. It is evident that the security of the users of such a system would be more easily granted if the server was able to carry out the task without getting any knowledge about the data provided by the users (not even the final result). Similarly the service provider may desire to keep the algorithms he is using to process the data secret, since they represent the basis for the service he is providing. Clearly the possibility of such kind of computation would be of invaluable help in situations like those described above.

Without being too specific and avoiding to discuss the details required by a precise definition, we may define *secure signal processing* or *signal processing in the encrypted domain* a collection of techniques that permit to solve the following problem: given the signals x_1 and x_2 signals (or data) belonging to Alice and Bob, compute the output of known function $f(x_1, x_2)$ without that Alice (Bob) gets any information about $x_2(x_1)$ in addition to that inferable from the output of the computation itself. As generalization it is possible to consider the case in which $f(.)$ is known only to a party and it has to be kept secret as well.

The number of possible applications of these techniques is virtually endless. Among the most interesting scenarios investigated so far we mention: private database access [1], in which the Alice accesses a server owned by Bob by means of an encrypted query; private data mining [17], in which two or more parties wish to extract aggregate information from a dataset formed by the union of their private data; secure processing of biometric data [8], in which biometric signals are processed in the encrypted domain to protect the privacy of the owners; watermarking of encrypted signals [16], for digital rights management within buyer-seller protocols; recommender systems [3], in which user's data is analyzed without disclosing it; privacy-preserving processing of medical data [4], in which sensitive medical data is processed by a non-trusted party, for remote medical diagnosis or any other form of homecare system whereby health conditions are monitored remotely.

7.3 Notation and Preliminaries

In the last few years, new techniques related to homomorphic encryption and multiparty computation showed that it is possible to perform several kinds of computations directly in the encrypted domain in an efficient way and without revealing the information hidden inside the cryptogram [2]. Following this direction, researchers developed many protocols where the protection of the inputs provided by the various parties involved in the computation is a crucial goal. The present work is part of this research streamline.

In the rest of the paper we will use the following notation:

- $\mathbb{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ is a set of m vectors $\in \mathbb{R}^n$
- With $\langle \cdot, \cdot \rangle$ we indicate the inner product: $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i b_i$
- With $[\![a]\!]$ we indicate the Paillier [20] encryption of a; if $\mathbf{a}$ is a vector we always indicate with $[\![\mathbf{a}]\!]$ the componentwise encryption of $\mathbf{a}$
- s is the cryptosystem security parameter (i.e. for short term security 1024 bit) and ℓ is the bit size of a cryptogram,[1] moreover $\mathbb{Z}_N$ is the ring in which the cryptosystem is defined ($s = \lceil \log_2 N \rceil$).

[1] Using the Paillier cryptosystem we have the following equality: $\ell = 2s$.

We recall that the following basic mapping[2] holds for Paillier's cryptosystem: $[\![x]\!][\![y]\!] = [\![x+y]\!]$ and $[\![x]\!]^y = [\![xy]\!]$.

Moreover, we recall the Big-$\mathcal{O}$ notation [15] that measures the computational complexity in bit operations. Assuming that the biggest number involved in the computation has ℓ bits we have $add = \mathcal{O}(\ell)$ to compute addition or $mult = \mathcal{O}(\ell^2)$ to compute multiplication and finally $exp = \mathcal{O}(\ell^3)$ to compute exponentiation. In the rest of this paper we often need to compute exponentiation by -1 (or negative numbers), this operation is equivalent to compute the multiplicative inverse in the space of the ciphertexts (namely $\mathbb{Z}^*_{N^2}$), this operation can be computed by using the extended GCD and its computational complexity is equal to compute an exponentiation, so $\mathcal{O}(\ell^3)$. Furthermore, we remind that for Paillier cryptosystem $enc \approx dec = \mathcal{O}(\ell^3)$.

7.4 Basic Building Blocks

In this section we introduce some basic building boxes that we will use to construct our protocol.

7.4.1 eMul

The first sub-protocol, `eMul` allows to compute the product of two Paillier ciphertexts obtaining $[\![xy]\!] = \texttt{eMUI}([\![\texttt{x}]\!], [\![\texttt{y}]\!])$ and is a well-known technique. Let us recall it. Suppose that Bob owns $[\![x]\!]$and$[\![y]\!]$encrypted with the public key of Alice, he can obfuscate both cryptograms adding two random numbers due to homomorphic additive properties and obtain $[\![x + r_x]\!]$ and $[\![y + r_y]\!]$. Now he sends these cryptograms to Alice, she decrypts and multiplies them finding: $w = xy + xr_y + yr_x + r_x r_y$, she encrypts it and sends back to Bob that computes:

[2] To be more precise, we have that given an instance of a Paillier cryptosystem and defined as $\mathcal{E}$ and $\mathcal{D}$ the functionalities of encryption and decryption respectively, the following properties hold:

$$\mathcal{D}(\mathcal{E}(x)\mathcal{E}(y)) = \mathcal{D}(\mathcal{E}(x+y))$$

and

$$\mathcal{D}(\mathcal{E}(x)^k) = \mathcal{D}(\mathcal{E}(kx)).$$

$$\begin{aligned}\llbracket w \rrbracket \llbracket x \rrbracket^{-r_y} \llbracket y \rrbracket^{-r_x} \left[\!\left[r_x r_y \right]\!\right]^{-1} &= \llbracket w \rrbracket \llbracket -x r_y \rrbracket \llbracket -y r_x \rrbracket \llbracket -r_x r_y \rrbracket \\ &= \llbracket w - x r_y - y r_x - r_x r_y \rrbracket \\ &= \left[\!\!\left[\underbrace{xy + x r_y + y r_x + r_x r_y}_{w} - x r_y - y r_x - r_x r_y \right]\!\!\right] \\ &= \llbracket xy \rrbracket\end{aligned} \tag{7.1}$$

obtaining exactly the product of the two encryptions.

Computing `eMul` requires two rounds (one from Bob to send the obfuscated ciphertexts and one from Alice to send back the result) and a bandwidth of 3ℓ (three ciphertexts are sent) with a computational complexity equal to: 3 *exp* needed to compute $\llbracket x \rrbracket^{-r_y}, \llbracket y \rrbracket^{-r_x}$ and $\left[\!\left[r_x r_y \right]\!\right]^{-1}$; 5 *mult* needed to obfuscate $\llbracket x \rrbracket, \llbracket y \rrbracket$ and to compute the additions to $\llbracket w \rrbracket$; 2 *dec* to obtain in plain $x + r_x$ and $y + r_y$ and finally 1 *enc* to encrypt the result, for a total asymptotic number of 6 *exp* operations. Later in this paper we refer to `eMul` using the following notation: $\texttt{eMul}(\llbracket \texttt{x} \rrbracket, \llbracket \texttt{y} \rrbracket) = \llbracket \texttt{x} \rrbracket \bullet \llbracket \texttt{y} \rrbracket$.

7.4.2 eInv

To realize our construction we will use another building block: `eInv`. This subprotocol works as follow: given an encrypted value $\llbracket x \rrbracket$ we have:

$$\texttt{eInv}(\llbracket x \rrbracket) = \left[\!\!\left[\frac{1}{x} \right]\!\!\right]. \tag{7.2}$$

To reach this goal we use a multiplicative blinding approach [14], in fact assuming T sufficiently bigger than x the multiplicative blinding Tx ca be assumed to be secure.[3] By this, Bob can compute $\llbracket Tx \rrbracket = \llbracket x \rrbracket^T$ by homomorphic properties and send the result to Alice that is able to decrypt obtaining Tx. Now, she computes $\frac{1}{Tx}$ encrypts it and sends back to Bob $\left[\!\!\left[\frac{1}{Tx} \right]\!\!\right]$. Bob removes the multiplicative blinding due to homomorphic properties: $\left[\!\!\left[\frac{1}{x} \right]\!\!\right] = \left[\!\!\left[\frac{1}{Tx} \right]\!\!\right]^T$ and obtain the desired result.

Computing `eInv` requires two rounds and a bandwidth of 2ℓ because only two cryptograms are sent: one from Bob and one from Alice. The computational complexity can be measured as: 1 *exp* for the multiplicative blinding, 1 *dec* for decryption, 1 *enc* for encryption and 1 *exp* to remove the blinding; for a total of 4 *exp* bit operations.

[3] Respect to the additive blinding, the multiplicative one requires a larger number of bits to achieve the same security level.

7.4.3 eDot

Another basic building block our protocol relies on is the the inner product between two encrytped vectors. More formally: given $[\![\mathbf{x}]\!]$ and $[\![\mathbf{y}]\!]$ encrypted with Alice's public key, the protocol $\texttt{eDot}([\![\mathbf{x}]\!], [\![\mathbf{y}]\!])$ computes $[\![\langle \mathbf{x}, \mathbf{y} \rangle]\!]$. To realize this sub-protocol it is possible to use data obfuscation. Given two vectors of random values $\mathbf{r_x}$ and $\mathbf{r_y}$, generated by Bob, he is able to evaluate the obfuscation of $[\![\mathbf{x}]\!]$ and $[\![\mathbf{y}]\!]$ as the componentwise product: $[\![\mathbf{x}]\!][\![\mathbf{r_x}]\!] = [\![\mathbf{x} + \mathbf{r_x}]\!]$ and $[\![\mathbf{y}]\!][\![\mathbf{r_y}]\!] = [\![\mathbf{y} + \mathbf{r_y}]\!]$. At this point Bob can send to Alice these two vectors of obfuscated values. Alice decrypts them and computes:

$$\begin{aligned}\langle \mathbf{x} + \mathbf{r_x}, \mathbf{y} + \mathbf{r_y} \rangle &= \sum_{i=1}^{n} (x_i + r_{xi})(y_i + r_{y_i}) \\ &= \sum_{i=1}^{n} x_i y_i + x_i r_{y_i} + y_i r_{xi} + r_{xi} r_{y_i} \\ &= \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} x_i r_{y_i} + \sum_{i=1}^{n} y_i r_{xi} + \sum_{i=1}^{n} r_{xi} r_{y_i}\end{aligned} \tag{7.3}$$

encrypts the scalar product obtaining: $[\![\langle \mathbf{x} + \mathbf{r_x}, \mathbf{y} + \mathbf{r_y} \rangle]\!] = [\![\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} x_i r_{y_i} + \sum_{i=1}^{n} y_i r_{xi} + \sum_{i=1}^{n} r_{xi} r_{y_i}]\!]$. Then she sends it back to Bob. Bob has to remove the obfuscation, to do this consider that:

$$\begin{aligned}&\left[\!\!\left[\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} x_i r_{y_i} + \sum_{i=1}^{n} y_i r_{xi} + \sum_{i=1}^{n} r_{xi} r_{y_i} \right]\!\!\right] \\ &= \left[\!\!\left[\sum_{i=1}^{n} x_i y_i \right]\!\!\right] \left[\!\!\left[\sum_{i=1}^{n} x_i r_{y_i} \right]\!\!\right] \left[\!\!\left[\sum_{i=1}^{n} y_i r_{xi} \right]\!\!\right] \left[\!\!\left[\sum_{i=1}^{n} r_{xi} r_{y_i} \right]\!\!\right]\end{aligned} \tag{7.4}$$

moreover Bob can compute:

$$\left[\!\!\left[\sum_{i=1}^{n} x_i r_{y_i} \right]\!\!\right] = \prod_{i=1}^{n} [\![x_i]\!]^{r_{y_i}} \tag{7.5}$$

$$\left[\!\!\left[\sum_{i=1}^{n} y_i r_{xi} \right]\!\!\right] = \prod_{i=1}^{n} [\![y_i]\!]^{r_{xi}} \tag{7.6}$$

$$\left[\!\!\left[\sum_{i=1}^{n} r_{xi} r_{y_i} \right]\!\!\right] \tag{7.7}$$

by using the additive property of the cryptosystem and the fact that he knows $\mathbf{r_x}$ and $\mathbf{r_y}$ in plain. Hence, Bob can compute:

Table 7.1 Sub-protocols complexities

Sub-protocol	Rounds	Bandwidth	No. of exponentiations
`eMul`	2	3ℓ	$6\,exp$
`eInv`	2	2ℓ	$4\,exp$
`eDot`	2	$(2n+1)\ell$	$(4n+2)\,exp$

$$[\![\langle \mathbf{x}, \mathbf{y} \rangle]\!] = [\![\langle \mathbf{x} + \mathbf{r_x}, \mathbf{y} + \mathbf{r_y} \rangle]\!] \left[\!\left[\sum_{i=1}^{n} x_i r_{y_i} \right]\!\right]^{-1} \left[\!\left[\sum_{i=1}^{n} y_i r_{xi} \right]\!\right]^{-1} \left[\!\left[\sum_{i=1}^{n} r_{xi} r_{y_i} \right]\!\right]^{-1}. \quad (7.8)$$

Computing `eDot` requires two rounds: one to send the obfuscated vectors to Alice and one to send back the result; with a bandwidth of $(2n+1)\ell$, because Bob sends two vectors of length n and Alice returns only one ciphertext. About the computational complexity we have:

$$\begin{aligned} (\underbrace{2n}_{\text{compute dot product}} + \underbrace{3}_{\text{remove obfuscation}})\, mult + 2n\, dec + 1\, enc + \underbrace{2n\, exp}_{\text{obfuscate vectors}} \\ \simeq\simeq (4n+2)\, exp. \end{aligned} \quad (7.9)$$

Table 7.1 shows the three complexities of the sub-protocols described so far.

7.5 Gram-Schmidt Orthogonalization on Encrypted Vectors

In the following section we introduce our construction to compute Gram-Schmidt orthogonalization on encrypted vectors. First of all we give a brief description of Gram-Schmidt process in its plain version, than we examine our privacy preserving protocol paying attention to security requirements and complexities.

7.5.1 Gram-Schmidt Orthogonalization in the Plain Domain

Gram-Schmidt Orthogonalization is a procedure for the orthogonalization of a set of vectors in a Euclidean space [23]. Given a set $\mathbb{V}$ of m vectors, it is possible to show that Algorithm 1 replaces $\mathbb{V}$ with a set of orthogonal vectors.

Algorithm 1 Gram – Schmidt Orthogonalization

1: $s_1 = \frac{1}{\langle \mathbf{v}_1, \mathbf{v}_1 \rangle}$
2: **for all** $i = 2$ to m **do**
3: **for all** $j = 1$ to $i-1$ **do**
4: $\mathbf{v}_i = \mathbf{v}_i - \langle \mathbf{v}_i, \mathbf{v}_j \rangle \; s_j \mathbf{v}_j$
5: **end for**
6: $s_i = \frac{1}{\langle \mathbf{v}_i, \mathbf{v}_i \rangle}$
7: **end for**

Note that the computational complexity is asymptotically equal to $\mathcal{O}(nm^2)$ multiplications [13]. Adding a normalization step at the end of Algorithm 1 it is possible to obtain an orthonormalized set. However, techniques for the normalization of encrypted vectors are out the scope of this paper.

7.5.2 Privacy Preserving Gram-Schmidt Protocol

We consider the case in which Bob owns a set of componentwise encrypted vectors with the public key of Alice and he needs to extract an orthogonalized version of them. For sake of simplicity we assume that $\mathbb{V}$ is a set of linearly independent vectors; this choice avoids the necessity of catching a *division by zero*. Otherwise, it is possible considering a variant where Bob asks to Alice to check if $\langle \mathbf{v}_i, \mathbf{v}_i \rangle$ is equal to zero, if is this the case Alice just sends back the encryption of zero.[4] We already introduced all the basic blocks we will use (See Sect. 7.4), so we can translate Algorithm 1 in the following Protocol 2.

Protocol 2 Privacy Preserving Gram – Schmidt Orthogonalization

1: $[\![s_1]\!] = \mathsf{eInv}(\mathsf{eDot}([\![\mathbf{v}_1]\!], [\![\mathbf{v}_1]\!]))$
2: **for all** $i = 2$ to m **do**
3: **for all** $j = 1$ to $i - 1$ **do**
4: $[\![\mathbf{v}_i]\!] = [\![\mathbf{v}_i]\!]\mathsf{eDot}([\![\mathbf{v}_i]\!], [\![\mathbf{v}_j]\!]) \bullet [\![s_j]\!]^{-1} \bullet [\![\mathbf{v}_j]\!]$
5: **end for**
6: $[\![s_i]\!] = \mathsf{eInv}(\mathsf{eDot}([\![\mathbf{v}_i]\!], [\![\mathbf{v}_i]\!]))$
7: **end for**

During Step 1 we use sub-protocols `eInv` and `eDot` to compute $s_1 = \frac{1}{\langle \mathbf{v}_1, \mathbf{v}_1 \rangle}$ used later in Step 4 to scale the projection. The main part is Step 4 where using sub-protocols `eDot` and `eMul` it is possible calculate $\mathbf{v}_i = \mathbf{v}_i - \langle \mathbf{v}_i, \mathbf{v}_j \rangle s_j \mathbf{v}_j$.

Security Discussion

To discuss the security of our construction we simply recall that in each step the data are encrypted when used by Bob or obfuscated, when used by Alice. Thus the privacy is achieved in the *honest but curious* model [10] due to IND-CPA property of Paillier cryptosystem and the security of obfuscation.

[4] It is simple to note that this reveals to Alice the rank of the set $\mathbb{V}$.

Complexities

We now briefly discuss the complexity of the proposed protocol. We assume that Bob already owns the vectors so in our complexity evaluation we do not consider: the bandwidth, the rounds and the operations needed to manipulate the vectors until the beginning of this protocol. We will examine the two principal steps in the protocol: Step 4and Step 6 (this is equal to Step 1). During *Step 4* there are $2n+1$ calls to `eMul` and 1 to `eDot`, so $4n+3$ rounds are needed with a bandwidth of $(2n+7)\ell$. Finally, resulting in a computational complexity of $(16n+8)\,exp$. Now, consider that it is necessary to execute Step 4 $\frac{m(m+1)}{2}$ times. Summarizing Step 4 requires: $\frac{m(m+1)(4n+3)}{2}$ rounds; a bandwidth of $\frac{m(m+1)(2n+7)}{2}\ell$ and a computational complexity of: $\frac{m(m+1)(16n+8)}{2}\,exp$. Finally, *Step 6* requires m executions and for each of them is just one call to `eInv` and one to `eDot`, so we can affirm that: $2m+2m=4m$ rounds are needed with a bandwidth[5] $(2+n+1)m\ell=(3+n)m\ell$ and a computational complexity $(4+4n+2)\,exp=(6+4n)m\,exp$. Summarizing we have:

$$\underbrace{\frac{m(m+1)(4n+3)}{2}}_{\text{Step 4}}+\underbrace{4m}_{\text{Step 6}}\simeq\mathcal{O}(nm^2) \tag{7.10}$$

rounds, for a bandwidth of:

$$\left(\underbrace{\frac{m(m+1)(2n+7)}{2}}_{\text{Step 4}}+\underbrace{(3+n)m}_{\text{Step6}}\right)\ell\simeq\mathcal{O}(nm^2\ell) \tag{7.11}$$

bits, and eventually:

$$\left(\underbrace{\frac{m(m+1)(16n+8)}{2}}_{\text{Step 4}}+\underbrace{(6+4n)m}_{\text{Step 6}}\right)exp=\mathcal{O}(nm^2\ell^3) \tag{7.12}$$

bit operations.

[5] Consider that `eDot` is computed on the same vector $\mathbf{v}_i$, so just n encryptions are sent by Bob instead than $2n$.

7.6 Conclusions

In this paper, we have proposed a secure protocol to compute the Gram-Schmidt orthogonalization on vectors encrypted with an additive homomorphic cryptosystem like Paillier's. We proved that our construction is secure because nothing is revealed to the parties. The idea was to propose a building block that could be used *off the shelf* in more complex privacy-preserving systems. To achieve our goal, we proved the protocol to be secure in the *honest but curious* model. Moreover, we show all the complexities involved: bandwidth, rounds and bit operations.

In the future, various improvements can be investigated, for instance, it is clear that for a real use of this protocol it is necessary to quantize the vectors because cryptosystems work on integer number representation, so a study of the error introduced by quantization will be really needed for practical implementations. Furthermore, techniques to normalize the encrypted vectors will be useful to generate an orthonormal version of the basis. Finally, it is possible to use techniques like those proposed in [5] to study a packetized version of our construction that could be much more efficient.

Acknowledgements This work is partially sponsored by MIUR under project Priv-Ware (contract no. 2007JXH7ET).

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. ACM. Sigmod. Rec. **29**(2), 439–450 (2000)
2. Ahituv. N., Lapid, Y., Neumann. S.: Processing encrypted data. Commun. ACM. **30**(9), 780 (1987)
3. Aimeur, E., Brassard, G., Fernandez, J.M., Onana, F.S.M., Rakowski, Z.: Experimental demonstration of a hybrid privacy-preserving recommender system. In: The Third International Conference on Availability, Reliability and Security. IEEE, pp. 161–170, (2008)
4. Barni, M., Failla, P., Kolensikov, V., Lazzeretti, R., Paus, A., Sadeghi, A., Schneider, T.: Efficient privacy-preserving classification of ECG signals. In: Workshop on Information Forensics and Security, WIFS (2009)
5. Bianchi, T., Piva, A., Barni, M.: Efficient pointwise and blockwise encrypted operations. In: Proceedings of the 10th ACM Workshop on Multimedia and Security. ACM, pp. 85–90, (2008)
6. Bjorck, A.: Solving linear least squares problems by Gram-Schmidt orthogonalization. BIT. Numer. Math.**7**(1), 1–21 (1967)
7. Brickell, J., Porter, D.E., Shmatikov, V., Witchel, E.: Privacy-preserving remote diagnostics. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM. 507p (2007)
8. Bringer, J., Chabanne, H.: An authentication protocol with encrypted biometric data. In: Proceedings of the Cryptology in Africa 1st International Conference on Progress in Cryptology, pp. 109–124. Springer-Verlag (2008)
9. Chui, C.K., Quak, E.: Wavelets on a bounded interval. Numer. Methods. Approx. Theory. **9**(1), 53–57 (1992)

10. Cramer, R.: Introduction to Secure Computation. Lectures on Data Security, pp. 16–62 (1999)
11. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-preserving face recognition. In: Privacy Enhancing Technologies, pp. 235–253. Springer
12. Failla, P.: Heuristic search in encrypted graphs. Accepted at IARIA International Conference on Emerging Security Information, Systems and Technologies. SECURWARE 2010 (2010)
13. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press (1996)
14. Kerschbaum, F.: Practical privacy-preserving benchmarking. In: Proceedings of the 23rd IFIP International Information Security Conference, pp. 17–31. Springer, (2008)
15. Koblitz, N.: A Course in Number Theory and Cryptography. Springer (1994)
16. Lemma, A., Van Der Veen, M., Tulys, P., Kalker, A.: Homomorphic encryption for secure watermarking. WO Patent WO/2006/129,293, 2006
17. Lindell. Y., Pinkas, B.: Privacy preserving data mining. J. cryptol. **15**(3), 177–206 (2008)
18. Ma, Y.D., Qi, C.L., Qian, Z.B., Shi, F., Zhang, Z.F.: A novel image compression coding algorithm based on pulse-coupled neural network and Gram-Schmidt orthogonal base. Dianzi Xuebao (Acta Electron. Sinica). **34**(7), 1255–1259 (2006)
19. Orfanidis, S.J.,: Gram-Schmidt neural nets. Neural Comput. **2**(1), 116–126 (1990)
20. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. Advances in Cryptology EUROCRYPT, pp. 223–238 (1999)
21. Rivest, R.L., Adleman, L. Dertouzos, M.L.: On data banks and privacy homomorphisms. Found Secure Computation, pp. 169–178 (1978)
22. Sharma, A., Paliwal, K.K.: Fast principal component analysis using fixed-point algorithm. Pattern. Recognit. Lett. **28**(10), 1151–1155 (2007)
23. Trefethen, L.N., Bau, D.: Numerical Linear Algebra. Society for Industrial Mathematics, (1997)
24. Zheng, W., Zou, C., Zhao, L.: Real-time face recognition using Gram-Schmidt orthogonalization for LDA. Pattern. Recognit. **2**, 403–406 (2004)

Chapter 8
A Peer-to-Peer Secure VoIP Architecture

Simone Cirani, Riccardo Pecori and Luca Veltri

Abstract Voice over IP (VoIP) and multimedia real-time communications between two or more parties are widely used over the Internet. The Session Initiation Protocol (SIP) is the current signaling standard for such applications and allows users to establish and negotiate any end-to-end multimedia session. Unfortunately current SIP-based platforms use a centralized architecture where calls between User Agents (UAs) are routed based on static public-reachable proxy servers, suffering of well-known scalability and availability problems. Moreover, security is currently poorly implemented and, when supported, it usually relies on a third-party trust relationship or on a Public Key Infrastructure (PKI). In this work we propose a completely distributed P2P VoIP architecture where calls are routed relying on a Location Service implemented through a Distributed Hash Table (DHT). End-to-end security is also provided without the use of any centralized server or PKI. Secure media sessions are established and authenticated on the basis of previously established sessions or by simple peer's voice recognition. The proposed architecture has been also implemented and publicly released.

Keywords SIP · Security · Peer to peer · Location services · DHT

S. Cirani · R. Pecori · L. Veltri (✉)
Department of Information Engineering, University of Parma, Viale G.P. Usberti 181/A, Parma, Italy
e-mail: luca.veltri@unipr.it

S. Cirani
e-mail: simone.cirani@tlc.unipr.it

R. Pecori
e-mail: riccardo.pecori@tlc.unipr.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_8,

8.1 Introduction

Peer-to-peer (P2P) architectures have been getting very popular in the last years thanks to the great variety of services they can provide. When they were born, they were mainly deployed as a simple, decentralized, and scalable way to exchange files, but they have now become very popular also for a lot of different services, exploiting the possibility of sharing bandwidth, computing power, storage capacity, and other resources between peers.

Thanks to the large diffusion of broadband connections, Voice-over-IP (VoIP) technology has reached more and more success; with its versatility it can serve from simple audio/video communications between two entities within the same administrated IP network to conference scenarios amongst different domains. The Session Initiation Protocol (SIP) [1] is the current signaling standard for VoIP applications, implemented in almost all standard VoIP devices. SIP can be used to setup any multimedia real-time session between two or more endpoints.

Although two SIP User Agents (UAs) can communicate directly without any intervening SIP infrastructure, which is why the protocol is sometimes described as peer-to-peer, this approach is impractical for a public service. In fact, according to a pure peer-to-peer SIP scenario the caller should know how to contact the callee, that is, it has to know the callee's IP address and port number which the callee's UA is listening on. Since this information is usually unknown in advance, it is necessary to rely on additional network elements (i.e., proxy servers, redirect servers, registrar servers) that, according to the SIP architecture, provide all the functionalities to register UAs and to properly route SIP calls. This is how all current SIP-based VoIP platforms have been implemented and work.

Unfortunately proxy servers represent a single-point of failure, and make the overall SIP architecture suffer of well-known scalability and availability problems.

Security is also an open issue for the current VoIP solutions since it is still poorly implemented and, when supported, it usually relies on a third-party trust relationship (e.g. users' keys are maintained and distributed by servers) or on a Public Key Infrastructure (PKI) that, in turn, is still not widely implemented and supported, and suffers of scalability problems too.

For such reasons, we studied and propose a new architecture that, widely adopting a P2P paradigm, provides secure VoIP service in a completely distributed, scalable, and reliable manner. According to such an architecture, SIP calls are routed via a Distributed Hash Table (DHT) based P2P infrastructure [2, 3] allowing the two UAs to establish any multimedia session regardless of the current points of attachment of the UAs and the available SIP nodes. Multimedia sessions are end-to-end secured on the basis of a Security Association (SA) that the two peers share without the use of any intermediary node. Such a SA is dynamically built between peers through a new key agreement protocol based on the MIKEY [4] Diffie–Hellman exchange authenticated, in a ZRTP-like fashion [5], exploiting previous established session keys or voice recognition based on the vocal reading of an authenticating short string.

The rest of the paper is organized as follows. In Sect. 8.2 current VoIP architectures and signaling and security protocols are briefly summarized. In Sect. 8.3 we present our P2P secure VoIP proposal, accurately describing both the distributed architecture for call routing and the key agreement protocol used to secure end-to-end VoIP communications. Section 8.4 presents a possible implementation and finally in Sect. 8.5 we draw some conclusions and indicate further works.

8.2 Current VoIP Architectures and Protocols

The Session Initiation Protocol (SIP) [1] is the IETF standard signaling protocol defined for initiating, coordinating and tearing down any multimedia real-time communication session between two or more endpoints. Such endpoints are commonly referred to as SIP User Agents (UAs). According to SIP, in order to setup a multimedia session a caller UA sends an INVITE request to the callee UA, addressed by a SIP URI that may identify: (i) the callee, or (ii) the actual contact IP address and port where the callee UA can currently be found. Since the former mechanism does not require the caller to know the actual contact address of the callee UA, it is the only way currently implemented by VoIP systems. However, such a method requires a way to dynamically map a user URI to the actual contact address of one or more UAs where he can be reached. In the standard SIP architecture, this is achieved by SIP intermediate nodes (like Proxy or Redirect SIP servers) and by a proper registration mechanism through which the UAs update their contact addresses. This results into a call scheme referred to as SIP trapezoid and formed by the caller UA, an outbound proxy (optional), the destination proxy (which the callee is registered with), and the callee UA. Unfortunately, such an architecture is server-centric and suffers of well-known scalability and availability problems. In order to setup a session in a real P2P fashion, a fully distributed SIP architecture is needed. Within the IETF a specific IETF WG, named P2PSIP, has been started, aiming to develop a new protocol for establishing and managing sessions completely handled by peers. The current IETF proposal is a binary protocol named RELOAD [6]. Differently from RELOAD, in our work we considered and implemented a protocol completely based on SIP. Other examples of non-IETF P2P VoIP protocols have been proposed in literature. The most relevant example is Skype [7].

As far as a secure media session has to be established, the two peers also require to agree on protocols, encryption and authentication algorithms, and keys, used to secure media contents, e.g. through the Secure Real-time Transport Protocol (SRTP) [8]. Such an agreement is often referred to as a Secure Association (SA). SAs between two peers are usually the result of a dynamic process involving a key agreement protocol (e.g. Internet Key Exchange (IKE) or TLS Handshake protocol for Transport Layer Security (TLS)) that in turn uses some pre-shared secret or public key to authenticate the SA negotiation. The core aspect of a key agreement

protocol is the exchange of a master-key (or a pre-master key). For reasons of freshness and of Perfect Forward Secrecy (PFS) guarantee, Diffie–Hellman (DH) exchange is usually deployed for such an aim. Unfortunately, DH is vulnerable to the well-known Man-in-the-middle (MITM) attack, through which a third party is able to trick both peers forcing them to agree to two different keys shared with itself. In order to prevent such a type of attack some sort of authentication of exchanged messages is needed. This can be achieved through the use of a pre-shared secret, by means of private and public keys and digital signature, or through the use of other authentication mechanisms such as Short Authentication String (SAS) [5]. The agreement on shared keys is a very strong assumption when applied to a P2P scenario in which peers want to communicate with each other without any pre-established relationship. Moreover, Certification Authorities (CAs) and PKI, often used in conjunction with digital signatures, introduce a form of centralization that does not fit with the scalability claimed for a P2P architecture.

In the following some current key agreement protocols are briefly summarized. Multimedia Internet KEYing (MIKEY) [4] is a key exchange protocol ancillary to SRTP, or other session-level security protocols, as it provides the means for setting up session keys. It can work in three different modes in order to generate a common master key (called TGK—Traffic-encrypting Generation Key) between two parties:

- Pre-shared key with key transport; it demands that an individual key is previously shared with every other peer
- Public key with key transport; it needs the knowledge of the responder's public key or of its certificate and the use of a centralized PKI
- Public key with authenticated (signed) Diffie–Hellman (DH) key exchange; it is more computationally expensive but grants perfect forward secrecy.

The main advantage of MIKEY, that is also the reason why we decided to use it, is that it is independent from any other media or signaling protocol, and can be encapsulated as part of the SDP payload during the session setup phase in SIP. Thus, it requires no extra communication overhead. In MIKEY the joint DH value is used directly as the derived key. Unfortunately, this leads to a key that does not appear as randomly generated, as it would be expected for a robust master key [9]. In our proposal we derive the master key from a hashed value of the DH value and previous master secrets, ensuring both random-likeness and dependance on previous secrets.

ZRTP [5] is an Internet-draft that describes a method to establish a session key for SRTP sessions using authenticated DH key exchange and encapsulating the relative messages in band in the media stream. The main feature of ZRTP is that the DH key exchange is authenticated through the possession of pre-shared secrets or the reading aloud of a SAS. As the authentication is not based on a centralized PKI infrastructure, it is particularly suitable for a pure P2P scenario as considered in this work.

8.3 P2P Secure VoIP Architecture

In this section we present a P2P VoIP architecture that can be used to setup a secure media session between two VoIP UAs without requiring prior knowledge of the callee's IP address, and without relying on any centralized, server-based infrastructure. In order to achieve such an architecture, two main components are needed:

- A method for routing calls and for performing session setup in a completely distributed, server-free, and reliable manner
- A method for establishing a SA and for agreeing on a session key, hopefully guaranteeing perfect forward secrecy (PFS).

The next two subsections detail how these two components are designed in our architecture.

8.3.1 Distributed Location Service

When a caller UA wants to initiate a session with a callee UA, it needs a way to obtain the actual network address (IP address and port) of the callee.

In the standard SIP architecture, this is usually achieved through the SIP trapezoid scheme. The goal of our architecture is basically to collapse the SIP trapezoid into a single line, connecting UAs directly. In order to create a fully distributed architecture for VoIP applications, we envisaged that SIP URI resolution can be provided by a peer-to-peer network, allowing for storing and lookup operations on SIP URIs. The most suitable peer-to-peer network type to do this is represented by Distributed Hash Tables (DHTs). DHTs are structured peer-to-peer networks which provide an information storage and retrieval service among a number of collaborating nodes. Information are stored as key/value pairs, as in regular hash tables. The structured nature of DHTs allows for upper-bounded (logarithmic) lookup procedures, which let DHTs scale well for high number of participating nodes. Based on DHTs, we have created a framework for a Distributed Location Service (DLS) [2]. The DLS is a peer-to-peer service built upon a DHT which allows to store and retrieve information about the location of resources in order to allow direct connections among the endpoints of a communication. From an application perspective, the DLS offers two main methods: put(key, value) used to store a mapping into the DHT, and get(key) to retrieve the value associated with the given key.

According to our P2P VoIP architecture, the DLS stores mappings between a URI identifying the resource (the callee UA) and a set of contact for the resource (where and how the UA is currently reachable). Such information includes the routable URL of the UA (containing IP address and port number), an optional human-readable display name, an expiration time, and an access priority value.

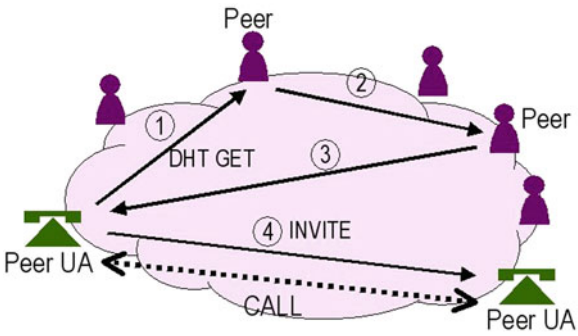


Fig. 8.1 P2P session setup through DHT-based DLS

An example of session setup between two SIP UAs through DLS is depicted in Fig. 8.1.

8.3.2 P2P VoIP Security

In this section a new key agreement protocol for multimedia P2P communications is described. The objective of the proposed protocol is to securely establish a master key between two multimedia SIP UAs that may or may not have already communicated with each other. Our proposal has been designed in such a way that it does not rely on any centralized PKI, as the new master key, created through a DH exchange, is authenticated in one of the following methods:

1. By means of the previously established secrets between the two peers
2. By means of a pre-shared key (PSK) or passphrase
3. By performing a ZRTP-like SAS based authentication.

Note that the latter method, which directly involves the two users by requiring them to read and compare the SAS (and verify the correctness of the voice of the remote peer), is used only in case the previous methods are not available or have failed.

The proposed mechanism is similar to the one used by the ZRTP protocol [5]; however the two mechanisms differ in some aspects and particularly:

- In the information exchanged during the key setup
- In the way such information is effectively encapsulated and exchanged
- In the possibility of authenticating the DH exchange through the use of a pre-shared secret (e.g. a passphrase).

Particularly, ZRTP establishes a new master key directly at media level, by using the RTP protocol as transport support for the key negotiation. Instead, the proposed solution uses MIKEY as negotiation protocol, opportunely encapsulated within SIP messages used for the session setup. In order to support a fully authenticated DH exchange, the MIKEY protocol has been extended to consider a

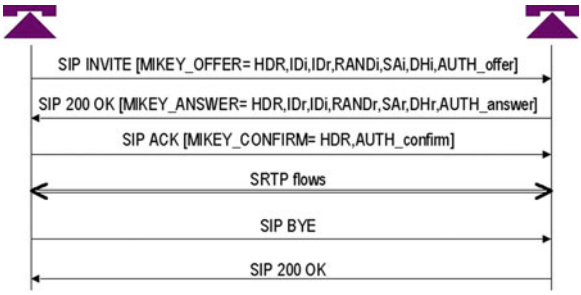


Fig. 8.2 Proposed three-way key agreement and session setup

MIKEY 3-way handshake (MIKEY originally supported DH in a 2-way request/response transaction). The new offer/answer/confirm handshake between the initiator (the caller) and the responder (the callee) is depicted in Fig. 8.2.

The Initiator is the entity sending a MIKEY OFFER to a Responder. The Offer encompasses: (i) a MIKEY header (HDR), (ii) identities of both the initiator (ID_i) and the responder (ID_r), (iii) a random value $(RAND_i)$, (iv) the list of the offered encryption and hash algorithms (SA_i), (v) the DH part of the initiator (DH_i), (vi) secrets used for the further exchange authentication $(AUTH_{offer})$. The $AUTH_{offer}$ is particularized depending on the selected authentication method. In case master keys from previously established sessions are used for authentication (1), the $AUTH_{offer}$ is formed by two values RS_1 and RS_2 (retained secrets) respectively obtained directly by the last two previously established master keys MK_1 and MK_2, as follows:

$$RS_j = HMAC(MK_j, \text{``Retained Secret''}), \quad j \in \{1, 2\}.$$

Such RS_j are used when computing the actual authentication field in the ANSWER and CONFIRM messages. The reason for sending two retained secrets is to face the case when in a previous setup, one of the two parties succeeded into computing the correct master key and the other one did not (e.g. caused by a fatal interruption during the session set up).

In case a pre-shared key (PSK) or passphrase is used (2), or in case of SAS based authentication (3), the $AUTH_{offer}$ field contains no data, and it is left empty. Note that authentication method 2 is used only if no previous secrets have been established and saved, while method 3 is used only if no previous secrets nor pre-shared secret is available between the two parties, or if the previous methods have failed.

Once the responder receives the MIKEY OFFER message, it controls the entire message, chooses $RAND_r$, SA_r, and its part DH_r of the DH exchange and calculates the hash (DH_{res}) of the new generated DH secret. He then generates the new master secret MK_0 as follows:

$$MK_0 = hash(DH_{res} \| ID_r \| RAND_i \| RAND_r \| MK_j)$$

where MK_j is the previous master key corresponding to the more recent RS_j that matches one of the local stored retained secrets; if both given retained secrets do not match the locally stored ones, or no RSs have been given at all, MK_j is left empty. The responder composes a MIKEY ANSWER message including: MIKEY HDR, the identities ID_i and ID_r, the random value $RAND_r$, the selected SA_r, the responder's DH part DH_r. Then, he calculates an authentication MAC of the entire MIKEY ANSWER message as follows:

$$HMAC_r = HMAC(MK_0, \text{MIKEY ANSWER})$$

and uses it to build an $AUTH_{answer}$ field that is appended to the ANSWER. The $AUTH_{answer}$ depends on the type of authentication that is performed:

- If $AUTH_{offer}$ contained retained secrets, the $AUTH_{answer}$ includes the matching RS_j and a $HMAC_r$
- If $AUTH_{offer}$ was empty, if a pre-shared key PSK is available, the responder composes an $AUTH_{answer}$ formed by the $HMAC_r$ encrypted with the PSK
- Otherwise $AUTH_{answer}$ is simply formed by the $HMAC_r$, and SAS authentication is performed successively.

Once the initiator receives the MIKEY ANSWER message, it checks the correctness of the $HMAC_r$ and, if it succeeds, it sends a MIKEY CONFIRM message including a $AUTH_{confirm}$ built with the same rules used by the responder. The $AUTH_{confirm}$, in turn, includes a $HMAC_i$ that authenticate the original MIKEY OFFER with the new master key, calculated as follows:

$$HMAC_i = HMAC(MK_0, \text{MIKEY OFFER})$$

If SAS authentication is required, it takes place after the multimedia session has been setup. A short authentication string is generated as follows:

$$SAS = \text{hexadecimal values of first } n \text{ bytes of } HMAC(MK_0, \text{``SAS''})$$

Both users are then invited to read the SAS aloud. If the SAS showed at both UA sides vocally matches, the new master key is considered secure and is saved in order to be used for authenticating successive master keys.

The described MIKEY offer/answer/confirm exchange is encapsulated within the standard SIP session setup procedure and can be used for establishing any P2P multimedia communication.

As an example, in Fig. 8.3 an INVITE message with MIKEY offer is shown.

8.4 Implementation

The proposed P2P Secure VoIP architecture has been completely implemented in Java language, according to the specifications provided in the previous section, and integrated into an open source SIP UA. For this purpose, we have implemented a

Fig. 8.3 SIP INVITE message including MIKEY offer

```
INVITE sip:bob@192.168.1.8:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.66:5070;port;branch=z9hG4bK1ef3d084
To: "Bob" <sip:bob@192.168.1.8:5080>
From: "Alice" <sip:alice@wonderland.net>;tag=857919546037
Call-ID: 747548207772@192.168.1.66
CSeq: 1 INVITE
Contact: <sip:alice@192.168.1.66:5070>; expires=3600
Security-Association: mikey
offer="AQQFgJraE7sDAAEAAAYT6bW4NALuaygB7msoAQPptbg0AAAGEwsAzdJQc
Gi0OVgGFFx3UOLse/3zehlCOFHvTuTZDQJhBgAAF0FsaWNIQHN0dWRlQHN0dWRlb
nRpLnVuaXByLmI0CgAAFUJvYkBzdHVkZW50aS51bmlwci5pdAMBAAAbAQMBAQECA
QIDBQMDAwMDBAIEBAUGBQUFBQUFDQAAzzhvYedNFRzEykg2Q856fxw6edNp5if+V
RthEancWoRwFk6++z7EAoMqrcU7Rn0t8ZiQX/aSBwQ+32rQhBhrm5W5modA0sQOh
iWikvrKxkMbv2m7rt61YTWqJpPzqhAnHeoLCsR7ZDo1EoMdREiJolRf2lcMX5Y+
9o3vdSLXFkSIN+IqeKTpE6yYjhvoTuN/I+3QcN6j0OWXKaW3eYR5PhKb55V+Tkrc
CXYehDpETyG6O55xO7G83p0WJBPeXBTAADN0vNUDuDlq9XkNAlONjER1eQ0CU42M
RHV5DQJTjYxEdXkNAIONjER"
Content-Length: 143
Content-Type: application/sdp

v=0
o=alice 0 0 IN IP4 192.168.1.66
s=-
c=IN IP4 192.168.1.66
t=0 0
m=audio 3000 rtp/avp 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
```

DHT-based DLS, completely transparent to the particular DHT algorithm and RPC protocol used for DLS maintenance. Our current implementation uses Kademlia [10] as DHT algorithm and dSIP [11] as DLS signaling protocol [2]. Pure P2P SIP calls are performed by exploiting the DLS as a SIP LS. After the P2P UA has enrolled into the DHT, it stores within the DHT the binding of the current UA's address to the user SIP URI. When a UA wants to perform a SIP call, the peer performs a lookup to resolve the target user's address, retrieves its location, and sends the INVITE request to the UA. Legacy SIP UAs are also supported by using a special peer named SIP Adapter and acting as SIP Proxy server. Registration requests received by the SIP Adapter peer are translated to DHT PUT requests, having the effect of storing the UA's contact into the DHT. Outgoing INVITE requests are sent to the SIP Adapter peer that will perform the lookup on behalf of the UA and forward the request.

The implementation has been based on the open source MjSip stack [12] that is a complete Java-based implementation of the layered SIP stack architecture as defined by RFC 3261 [1], supporting both JavaSE and JavaME (J2ME/CLDC1.1/MIDP2.0). The key agreement protocol described in the previous section has been also implemented ([13] reports a first implementation of such a protocol) and integrated. According to that, when a UA contacts a remote UA for the first time no pre-stored keys are available and the media flows are established through SRTP by using a new unauthenticated DH generated master key. In such a case, the two parties perform SAS authentication, that in turn leads the two users to read a displayed SAS string. If the authentication succeeds, the new key is stored and re-used for authenticating further key agreement procedures, without requiring SAS re-authentication.

8.5 Conclusions

In this paper we have presented a distributed architecture for P2P secure session initiation. In order to correctly route calls between any peers, a Distributed Location Service has been considered, based on DHT.

Security of end-to-end sessions is guaranteed by media encryption and authentication via SRTP protocol. The SRTP master key is negotiated through a proper new key agreement protocol that does not require any third-party relationship. The key agreement is performed via the DH algorithm and authenticated through a previously used session key (between the two parties), if available, or by means of vocal reading and recognition of a short string (SAS).

The proposed architecture has been also implemented in Java language, based on a SIP open source implementation. The current implementation includes a DLS based on Kademlia as DHT algorithm and dSIP as communication protocol. We also realized the Peer Adapter (that is a peer that can act as a standard SIP proxy for legacy UAs) in order to route calls between DHT unaware UAs.

References

1. Rosenberg, J., et al.: RFC 3261: SIP: Session Initiation Protocol. IETF StandardTrack. http://www.ietf.org/rfc/rfc3261.txt (2002)
2. Cirani, S., Veltri, L.: Implementation of a framework for a DHT-based Distributed Location Service. In: Proceedings of the 16th International Conference on Software, Telecommunications and Computer Networks, Split–Dubrovnik, Croatia (2008)
3. Cirani, S., Veltri, L.: A Kademlia-based DHT for Resource Lookup in P2PSIP. IETF Internet-Draft ciranip2psip-dsip-dhtkademlia-00. http://tools.ietf.org/html/draft-cirani-p2psip-dsip-dhtkademlia-00 (2007)
4. Arkko, J., et al.: RFC 3830: MIKEY: Multimedia Internet KEYing. IETF Standard Track. http://tools.ietf.org/html/rfc3830 (2004)
5. Zimmermann, P., Johnston, A., Callas, J.: ZRTP: Media Path Key Agreement for Secure RTP. IETF Internet-Draft draft-zimmermann-avt-zrtp-21. http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-21 (2010)
6. Jennings, C., et al.: REsource LOcation And Discovery (RELOAD) Base Protocol. IETF Internet-Draft draft-ietf-p2psip-base-09. http://tools.ietf.org/html/draft-ietf-p2psip-base-09 (2010)
7. Baset, S.A., Schulzrinne, H.G.: An analysis of the Skype peer-to-peer internet telephony protocol. In: Proceedings of the 25th IEEE International Conference on Computer Communications, Barcelona, Spain (2006)
8. Baugher, M., et al.: RFC 3711: The Secure Real-time Transport Protocol (SRTP). IETF Standard Track. http://www.ietf.org/rfc/rfc3711.txt (2004)
9. Gupta, P., Shmatikov, V.: Security analysis of voice-over-IP protocols. In: Proceedings of the 20th IEEE Computer Security Foundations Symposium, Venice, Italy (2007)
10. Maymounkov, P., Mazires, D.: Kademlia: a peer-to-peer information system based on the XOR metric. In: 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA (2002)

11. Bryan, D.: dSIP: A P2P Approach to SIP Registration and Resource Location. IETF Internet-Draft draft-bryan-p2psip-dsip-00. http://www.p2psip.org/drafts/draft-bryan-p2psip-dsip-00.html (2007)
12. Veltri, L.: MjSIP Project. http://www.mjsip.org/ (2010)
13. Pecori, R., Veltri, L.: A key agreement protocol for P2P VoIP applications. In: Proceedings of the 17th International Conference on Software, Telecommunications and Computer Networks, Hvar–Korcula–Split, Croatia (2009)

Chapter 9
Improving QoS of Femtocells in Multi-operator Environments

Franco Mazzenga, Marco Petracca, Remo Pomposini and Francesco Vatalaro

Abstract The growth of self-installed femtocells in residential and office environments triggers harmful femto-to-femto interference levels. In order to overcome this problem, we suggested that operators mutually share their licensed spectrum allowing femtocells to exploit also the frequency resources of other operators. By assuming mutual arrangements among operators, we proposed algorithms enabling femtocells to dynamically select the best operating channel among those available from every operator just based on local interference measurements. In such a way the interference between femtocells belonging to the same operator can be considerably reduced. In this paper we describe and evaluate performance of the proposed dynamic frequency selection algorithms in terms of outage probability and average throughput per femtocell. In our analysis we examine various scenarios in which we consider different number of available frequency channels. Results show that in a multi-operator environments the proposed approach allows to improve QoS in femtocell networks.

Keywords Femto-cell · Quality of service · Spectrum sharing

F. Mazzenga · M. Petracca · R. Pomposini (✉) · F. Vatalaro
Department of Electronic Engineering, University of Rome "Tor Vergata",
v. del Politecnico 1, Rome, Italy
e-mail: pomposini@ing.uniroma2.it

F. Mazzenga
e-mail: mazzenga@ing.uniroma2.it

M. Petracca
e-mail: petracca@ing.uniroma2.it

F. Vatalaro
e-mail: vatalaro@ing.uniroma2.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_9,

9.1 Introduction

Femtocells are small domestic low cost and low power cellular-based access points, also known as Home Nodes B (HNB's) or "home base stations", which are self-installed by consumers and are remotely managed by operators. HNB's transmit with a range of tens of meters in licensed band (e.g., UMTS frequency bands), thus avoiding the need for dual mode devices, and provide mobile handsets with high data rate wireless access to the mobile operator network through broadband wired connection, such as cable, xDSL or optical fiber. The need for femtocells derives from the consideration that mobile terminals are predominantly used within closed spaces [2]. Indeed, since most of the mobile radio traffic is spent in the home and in workplaces, a better indoor coverage is wished in order to increase the available bit-rate as well as to off-load macrocells [1, 3]. In such a way, femtocells allow indoor mobile radio users to use advanced data services, such as high quality video and audio streaming, downloads, on-line gaming and other multimedia applications, with a higher efficiency in the use of spectrum resources.

However, in the next envisaged scenario where operators can assign dedicated, common or partially common channels to femtocell with respect to the frequencies allocated to the macrocell network, macro-to-femto (cross-tier) and/or femto-to-femto (co-tier) interference can occur [4, 9]. In particular, even when cross-tier interference is avoided by allocating separated channels, the self-installation nature of femtocells can lead to harmful co-tier interference especially in highly dense HNB's environments. This can impact on the QoS performance, up to compromise the opportunity of enjoying high data rate services.

For this reason, in [8] we proposed Dynamic Frequency Selection Algorithms (DFSA) aimed to redistribute the available spectrum belonging to different network operators among femtocells just based on local interference measurements. Under the assumption that the regulatory framework permits the mutual exchange of frequency bands among operators, HNB's can dynamically select the best operating frequency according to a Cognitive Radio (CR) concept. This solution allows to obtain improved performance in terms of network capacity and achievable throughput per femtocell with respect to the random frequency assignment resulting from the not coordinated deployment of HNB's.

In this paper we evaluate the performance of DFSA in terms of outage probability and average *Signal-to-Interference Ratio* (SIR) in different scenarios. Furthermore, two cases are analyzed: an ideal situation in which all the femtocells in the considered scenario adopt the proposed DFSA; a more realistic situation in which only some HNB's conform to DFSA.

The paper is organized as follows. In Sect. 9.2 we provide a brief overview of the regulatory aspects concerning the possibility of performing mutual frequency exchange among different femtocell operators. The description of the proposed algorithms is reported in Sect. 9.3. In Sect. 9.4 we detail the interference scenarios considered for the performance analysis. Simulation results are shown in Sect. 9.5. Finally, conclusions are drawn in Sect. 9.6.

9.2 Regulatory Aspects

The proposed solution to limit co-tier interference in femtocell networks is based on the assumption that network operators make arrangements one with each other (similar to roaming agreements) to allow the reciprocal exchange of operating frequency channels. However, while the infrastructure sharing among telecom service providers is a mandatory policy by the European Commission (EC) [5], the simultaneous mutual interchange of spectrum bands among network operators is not currently permitted. Nevertheless, the guidelines proposed by some regulatory bodies open interesting perspectives in this regard. Indeed, the policy programme for the use of the European Union's radio spectrum foresees an efficient and flexible spectrum management as well as the promotion of collective use of spectrum [7]. Moreover, the Radio Spectrum Policy Programme (RSPP) encourages the development of standards able to avoid harmful interference or disturbance by other radio or non-radio devices by means of efficient spectrum usage techniques, especially when high density of radio devices occurs [6].

In this perspective, the demonstration of the benefits deriving from the sharing of licensed frequency bands among operators can contribute to review the communications regulatory framework

9.3 Dynamic Frequency Selection Algorithms

We propose two different approaches for femtocells that differ in the preference or not to use the frequency band of their operators. In the first case, each femtocell takes into account its subscription to the specific operator and attempts to use its own band until the SIR is above the required threshold ρ_0. In the second case, the femtocells are absolutely *greedy* and aim to maximize their SIR careless of which operator offers the less interfered channel. It means that also if a femtocell measures a SIR enough to guarantee the desired QoS level on its frequency, it searches for another band that can maximize the throughput. We refer to the two algorithms as Greedy Dynamic Frequency Selection Algorithms (GDFSA) and Operator-oriented Dynamic Frequency Selection Algorithms (ODFSA), respectively.

For the sake of simplicity, here we describe the proposed DFSA in the case that each operator provides its femtocells with a single dedicated shared channel. However, the proposed algorithms can be easily extended to the case of more channels per operator assigned to femtocells.

In a multi-operator scenario with N network operators, the flow chart for modeling the ODFSA by femtocells transmitting at the maximum power level is reported in Fig. 9.1. A more detailed description of both the DSFA including the case of femtocells implementing power control mechanisms can be find in [8].

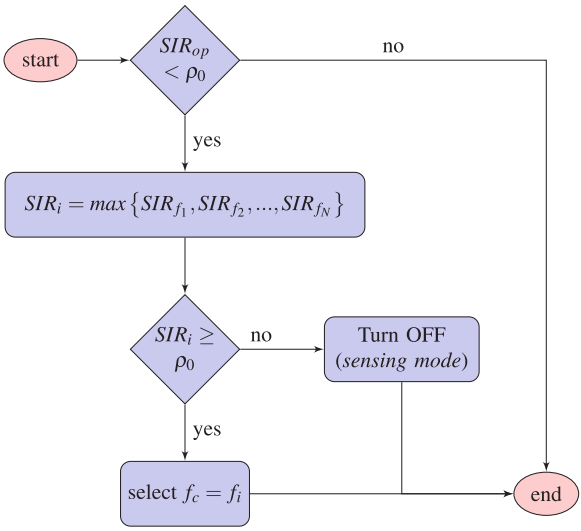


Fig. 9.1 Flow chart of the ODFS algorithm

Once the femtocells of the different operators are deployed by users in the area, they perform the start-up procedure [11], after which they start transmitting on their operating frequency. Since in the ODFSA each femtocell prefers to maintain the channel of its operator until the SIR on that channel (i.e. SIR_{op}) is above the required threshold ρ_0, if the measured SIR_{op} verifies this QoS condition, the femtocell uses the channel of its operator although other channels are less interfered and it enables its transmission state if it was temporarily off. Otherwise, the femtocell searches for the less interfered channel among those of the other operators. However, if also the SIR on the new selected channel does not match the QoS requirements, the femtocell sets its status to OFF, i.e. it just continues to perform spectrum sensing and interference measurements without accessing the channels (*sensing mode*). This programmed standby feature can be envisaged in order not to damage the other femtocells: since the interference conditions prevent the femtocell from transmitting, it temporarily disables transmission and switches into sensing mode. When the measured SIR_{op} or the maximum SIR among those of the other operators returns above the threshold, the femtocell can operate on the corresponding selected channel and enables transmission if previously it was off.

Note that the GDFSA can be seen as a special case of the ODFSA, since the only difference is that each femtocell immediately selects the channel that maximizes its SIR regardless of which operator has license for that frequency band. It means that the first conditional block is skipped in Fig. 9.1. This different strategy results in a maximization of the femtocell throughput with respect to the ODFSA.

It is worthwhile to note that the described cognitive algorithms are very simple and easy to implement.

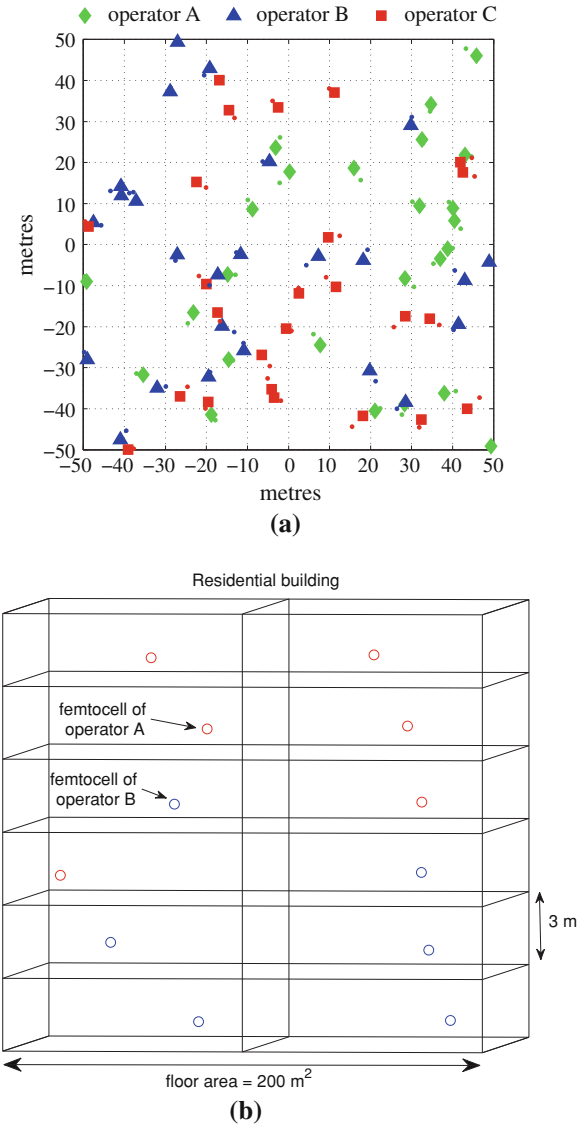


Fig. 9.2 Reference scenarios for the interference analysis. **a** Residential area, **b** Residential building

9.4 Scenarios Description

In Fig. 9.2 we show the reference scenarios considered for our analysis. Figure 9.2a represents a typical residential area of $100 \times 100\,\text{m}^2$ where femtocells belonging to different network operators are randomly positioned in accordance to an uniform spatial distribution. User terminals are assumed to be located 10 m away from their HNB's. The other assumed scenario is depicted in Fig. 9.2b, where we consider

femtocells installed in a six floor residential building. HNB's are indicated with circle and each colour represents the assignment to a different operator. As regards the number of femtocells within the building, we consider two cases corresponding to different levels of HNB's density. By assuming one HNB for each apartment, in the first case each floor has two apartments (low–medium density scenario), whereas in the second case four apartments are considered for each floor (high density scenario). According to the typical self-installation by users, HNB's are randomly located inside the apartments. We assume that the floor area is 200 m^2 and each floor is 3 m height. Equal-area apartments are assumed in which user terminals are located at a distance of 4 m from the HNB.

In both scenarios, a typical multi-operator environment is considered where each femtocell is subscribed to one of the N network operators. In our analysis we assume both $N = 2$ and $N = 3$ operators providing services in the considered area. The initial assignment of each femtocell to an operator (and therefore to an operating frequency) is randomly performed in accordance to an uniform distribution. We assume that each operator allocates one dedicated frequency band for femtocell communications. Hence HNB's subscribed to a same operator do not suffer from cross-tier interference, whereas they interfere one with each other due to the sharing of a single radio channel. We assume that all the femtocells transmit at the maximum power level allowed by the standard specification, i.e. $P_{tx} = P_{max} =$ 20 dBm [10].

We used different propagation models depending on the type of link considered. As regards the indoor path loss we assume the ITU-R P.1238 model [10], expressed as:

$$L_{IN}(d)[\text{dB}] = L_{50}(d) + L_{FM} + L_W \tag{9.1}$$

where L_{FM} is the additional shadow fade margin and L_W is the penetration loss related to the wall between adjacent apartments (i.e. apartments on the same floor) or the outer wall of the buildings. From [10], we assume $L_{FM} = 9.9$ dB and $L_{FM} = 7.7$ dB for the residential building and the residential area scenarios, respectively. $L_{50}(d)$ is the median path loss at a distance d, i.e. the loss exceeded at 50% of positions at that distance, given by the following expression:

$$L_{50}(d)[\text{dB}] = 20 \cdot \log_{10} f_c + 10 \cdot \gamma \cdot \log_{10} d + L_f(n_f) - 28 \tag{9.2}$$

In Eq. 9.2 f_c is the operating frequency, γ is the indoor path loss exponent and $L_f(n_f)$ is the floor penetration loss, which varies with the number of penetrated floors n_f. As recommended by the ITU-R [10], both γ and $L_f(n_f)$ depend on the operating frequency and the environment. We assume that femtocells operate in a residential environment at frequencies f_c around 1800 MHz, with a channel spacing between different network operators $\Delta f_c = 10$ MHz. Based on this assumption, from [10] we set $\gamma = 2.8$ and $L_f(n_f) = 15 + 4 \cdot (n_f - 1)$. In the building scenario we assume $n_f = 6$, while in the residential area scenario we consider ground level apartments ($n_f = 0$).

Note that for the residential building scenario the indoor path loss model is used for both useful signal and interfering signals by setting $L_W = 0$ dB and $L_W = 15$ dB, respectively. Conversely, in the residential area scenario only the attenuation of the useful signal is modeled with (9.1), thus assuming $L_W = 0$ dB. As for the outdoor attenuation model needed for assessing the interfering signal propagation in this latter scenario, we considered the following expression:

$$L_{OUT}(d)[\text{dB}] = \text{MCL} + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) + 2 \cdot L_W \tag{9.3}$$

where $d > 10$ m is the distance in metres from the considered femtocell, MCL is the minimum coupling loss in decibel for $d_0 = 1$ m, n is the outdoor path loss exponent assumed equal to 4. For the sake of simplicity we considered just two external walls crossed by the interfering signal and we set $L_W = 10$ dB.

In our analysis we focus on the following two different situations:

1. *On–off* case, i.e. all the femtocells in the area can or can not implement the proposed frequency selection algorithms.
2. *Hybrid* case, in which a certain variable percentage of HNB's conforms to DFSA, while the rest of the femtocells are unable to implement CR techniques.

The first case is merely aimed at evaluating the QoS improvement deriving from the proposed DFSA, while the second situation is considered to assess the impact on QoS of a percentage of HNB's defecting from the DFSA. This can be expected in a more realistic scenario where failures or a transient state towards the complete adoption of DFSA can occur. According to these considerations, we apply the first case to the residential area scenario, while in the building scenario the hybrid situation is considered.

9.5 Performance Results

We run simulations to evaluate the impact of the proposed DFSA on the QoS performance of femtocells in the considered scenarios. The outage probability and the average SIR per active femtocell are assessed as a function of the number of HNB's in the area and the percentage of non DFSA-conformed HNB's in the *on−off* and *hybrid* cases, respectively. Since the outage performance with GDFSA and ODFSA are very similar [8], in this work we report results obtained with GDFSA, which allows to maximize the average SIR. Hence in the following of this section we refer to the *greedy* version of the algorithm as DFSA. The required QoS threshold ρ_0 is assumed equal to 9.4 dB and 16.4 dB for the residential area and residential building scenarios, respectively. As in [8], for both scenarios each femtocell is randomly scheduled for the interference measurements and the status update. Furthermore, the simulation time is appropriately set to guarantee the convergence of the DFSA, i.e. at the end of the test each femtocell has selected the

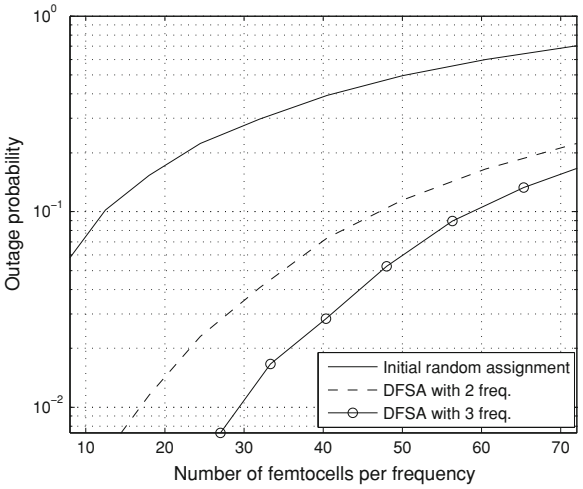


Fig. 9.3 Outage probability versus the number of femtocells

best operating frequency channel. Simulation results are obtained using a Monte Carlo based approach.

As for the first considered scenario, in Fig. 9.3 we report the outage probability obtained without the chance to exchange frequency channels among operators compared with the curves related to DFSA for two and three available channels. A marked improvement of network capacity is observed by applying the proposed algorithms with respect to the initial random frequency assignment. Moreover, note that an increase of the number of available frequency bands results in an improvement in the number of active femtocells per frequency. As an example, given an outage probability of 5%, in the case of two frequency channels, we obtain about 34 served femtocells per frequency, while in the same scenario with three network operators we have 47 active femtocells per channel. We also report in Fig. 9.4 the average achievable SIR per active femtocell as a function of the number of femtocell per frequency in the area. We can observe that the proposed algorithms perform considerably better than the initial random frequency assignment. The results show that similar values of SIR are obtained considering two and three available frequency channels (e.g., a SIR of about 13.6 dB is obtained for 100 and 150 femtocells in the area, respectively). This is due to the *sensing mode* feature of the proposed DFSA, which permits to preserve the throughput performance for the active femtocells, i.e. those HNB's which are not in outage (see Fig. 9.3).

As regards the hybrid scenario, for low–medium density of HNB's in the area (i.e. two femtocells per floor), when two operators provide services in the considered residential area, the complete defection from the DFSA causes an increase of the outage probability, which ranges from about 8.4% (which is equivalent on average to 1 outage femtocell in the building) when all the femtocells adopt the proposed algorithm up to about 50% in the all non DFSA-conformed HNB's case. This outstanding degradation is due to the impossibility to perform an efficient

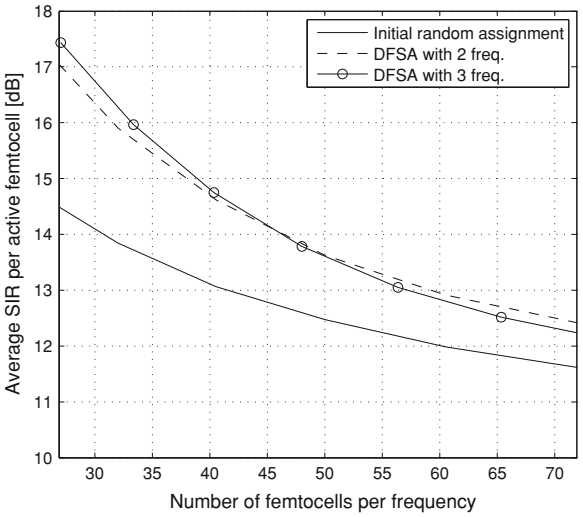


Fig. 9.4 Average SIR versus the number of femtocells

allocation of frequency channels by means of the proposed algorithms. The obtained results show similar trends also when three operators are considered. However, in this case the gap between an all DFSA-compliant scenario and the case of 100% non DFSA-conformed HNB's is larger than the two operators scenario. Indeed, when three operating frequencies are available for femtocells, a smart cognitive selection of the channel lead to a marked improvement of outage probability with respect to the initial random assignment.

The performance related to the outage probability are reflected in the average SIR experimented by femtocells which are not in outage, as shown in Fig. 9.4. As expected, for two available frequency bands with respect to the all DFSA-conformed HNB's case a slight worsening of average SIR is observed when all the femtocells do not implement the DFSA. This trend is reflected in Fig. 9.7, which reports the cumulative distribution functions (CDFs) of the SIR for different percentage of HNB's defecting from the DFSA. This results are obtained considering the SIR of all the femtocells in the area. We can note that in general it is better to adopt the DFSA, but with the increase of the percentage of non DFSA-conformed femtocells the probability of obtaining higher values of SIR increases. This is more evident when three operators provide services in the considered area, resulting in an average SIR difference of more than 2 dB between the selfish behaviour and the defection from DFSA. As shown in Fig. 9.4, when three operators provide services in the considered area it's much better for femtocells to conform to the proposed DFSA since the average SIR per HNB's increases of more than 2 dB.

In the high HNB's density case (i.e. four femtocells per floor), the implementation of DFSA always allows to maximize the network capacity and the achievable average SIR. In particular, starting from higher values of outage probability due to the increased number of femtocells per area, the consideration related to the trends of the outage curves are the same of the low–medium density scenario.

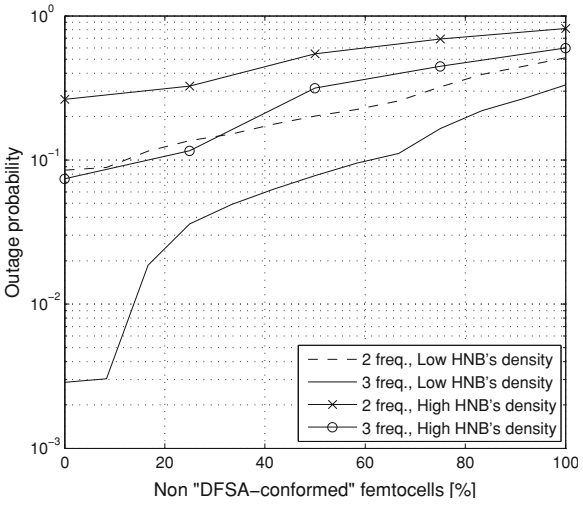


Fig. 9.5 Outage probability versus the percentage of non DFSA-conformed femtocells

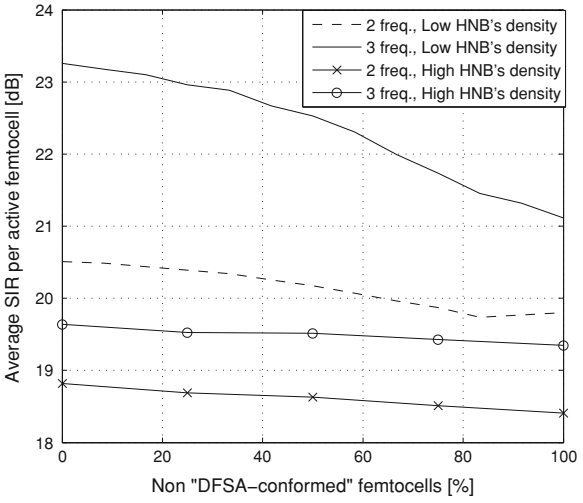


Fig. 9.6 Average SIR versus the percentage of non DFSA-conformed femtocells

As for the average SIR of femtocells in quality, we can observe a very slight performance decrease passing from the case of all DFSA-compliant HNB's to the situation of complete defection from the DFSA. This is due to the high density of femtocells in the area, which on one side limits the capability of DFSA to increase the throughput performance and on the other side statistically leads to a greater value of the average SIR of a few femtocells that are not in outage in the random frequency assignment (see Figs. 9.5 and 9.6). Indeed, in the random channel distribution single femtocells of one operator neighbour to cluster of femtocells belonging to the other operator can occur. This implies that a lot of femtocells measures $\mathrm{SIR} < \rho_0$, while a few "lucky" femtocells can experiment high values of SIR, as visible in the last curve of Fig. 9.7. Thus, by considering only the active

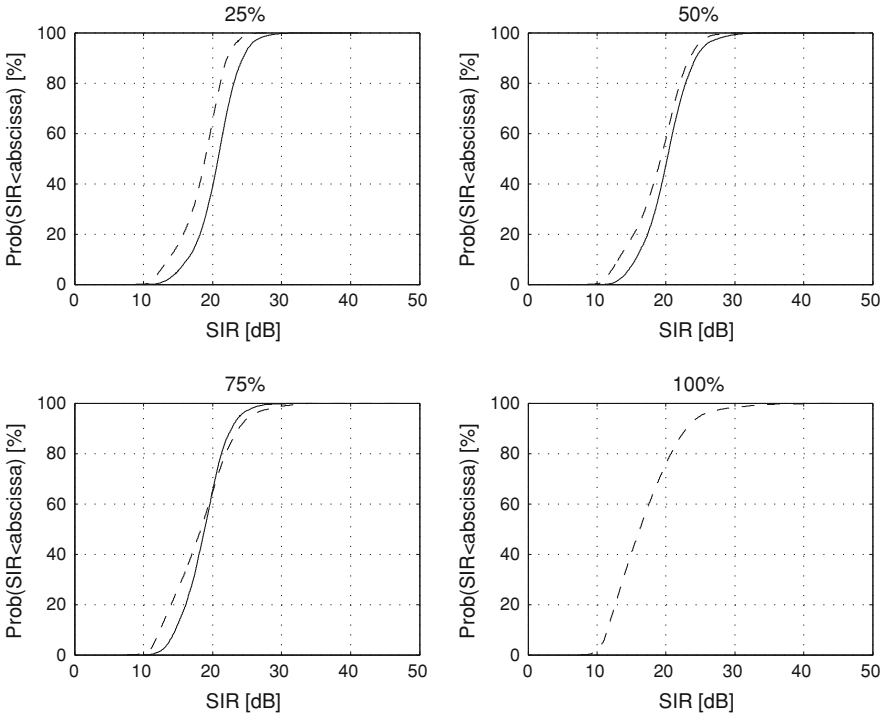


Fig. 9.7 CDF for various percentage of femtocells that do not implement the DFSA algorithm

femtocells we obtain a marginal SIR improvement with the implementation of DFSA.

In general, we can argue that in the case of femtocells defecting from DFSA the average SIR of femtocells which are not in outage shows a very slight decrease with respect to the best performance (i.e. the all DFSA-conformed HNB's case) to the detriment of marked worse outage probability. Thus, the best situation is verified when all the femtocells implement the proposed DFSA.

From the obtained results we can observe that the higher the density of femtocells in the area, the greater the number of operating channels needed for enjoying high data rate services and the more opportune is the implementation of the DFSA.

9.6 Conclusions

Harmful co-tier interference can occur due to the self-installation nature of femtocells especially in highly dense environments. We proposed distributed CR algorithms to enable each femtocell to dynamically select the operating frequency

among those available from every operator just based on local interference measurements.

In this paper we proved by results the marked improvement of QoS performance achievable with the implementation of DFSA. In particular, we evaluated by simulations the outage probability and the average SIR per femtocell. We assumed both cases of two and three operators providing services in different scenarios, where each operator allocates one dedicated frequency channel for femtocell communications. We can argue that in a multi-operator environments the proposed solution offers advantages to all operators in terms of achievable network capacity and average throughput provided to customers.

References

1. Calin, D., Clazaussen, H., Uzunalioglu, H.: On femto deployment architectures and macrocell offloading benefits in joint macro-femto deployments. IEEE Commun. Mag. **48**(1), 26–32 (2010)
2. Chandrasekhar, V., Andrews, J., Gatherer, A.: Femtocell networks: a survey. IEEE Commun. Mag. **46**(9), 59–67 (2008)
3. Claussen, H.: Performance of macro- and co-channel femtocells in a hierarchical cell structure. In: IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (2007)
4. de la Roche, G., Valcarce, A., Lopez-Perez, D., Zhang, J.: Access control mechanisms for femtocells. IEEE Commun. Mag. **48**(1), 33–39 (2010)
5. European Commission: Directive 2009/140/EC of the European parliament and of the council. Off. J. Eur. Union L **337**(52), 37–68 (2009)
6. European Commission: Radio spectrum policy: first programme. In: COM/2010/0471 final—COD 2010/0252 (2010)
7. European Commission: Spectrum: commission proposes to ensure availability of radio frequencies for new and faster wireless services. In: MEMO/10/425 (2010)
8. Mazzenga, F., Petracca, M., Pomposini, R., Vatalaro, F., Giuliano, R.: Algorithms for dynamic frequency selection for femto-cells of different operators. In: 21st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Istanbul (2010)
9. ping Yeh, S., Talwar, S., choon Lee, S., Kim, H.: WiMAX femtocells: a perspective on network architecture, capacity, and coverage. IEEE Commun. Mag. **46**(10), 5865 (2008)
10. Saunders, S., Carlaw, S., Giustina, A., Rai Bhat, R., Srinivasa Rao, V., Siegberg, R.: Femtocells: opportunities and challenges for business and technology. Wiley, Hoboken (2009)
11. Zhang, J., de la Roche, G.: Femtocells: technologies and deployment. Wiley, Hoboken (2010)

Chapter 10
Autonomic Network Configuration in IEEE 802.15.4: A Standard-Compliant Solution

Francesca Cuomo, Anna Abbagnale and Emanuele Cipollone

Abstract In the autonomic networking framework, particular attention deserves the application of this paradigm to Wireless Personal Area Networks (WPANs). In this context, network algorithms need to be adaptive, robust and scalable, with fully distributed and self-organizing architectures. We study automation and self-management of the IEEE 802.15.4 WPANs formation, with the aim of having robust and energy efficient topologies that can be used for low-rate, low-cost and low-power communications. Specific attention is given to the proposal of a standard-compliant procedure to reconfigure the network coordination. Beside the description of the proposed implementation issues, we provide a performance evaluation of the benefits, in terms of energy consumption, of our reconfiguration procedure.

Keywords IEEE 802.15.4 · ZigBee · Coordinator election · Wireless personal area networks

10.1 Introduction and Motivations

The evolution of Internet in the next years suggests that the development of self-managing, self-configuring and self-regulating network and communication infrastructures will be an area of considerable research and industrial interest [4]. In this context, IEEE 802.15.4 Wireless Personal Area Networks (WPANs) will be used in a large variety of applications, thanks to their *pervasive* and *autonomic* nature, and will be a part of the *FutureInternet* as for instance in case of the *Internet of Things*.

F. Cuomo (✉) A. Abbagnale · E. Cipollone
DIET, University of Rome "Sapienza", via Eudossiana 18, 00184 Rome, Italy
e-mail: francesca.cuomo@uniroma1.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_10,

IEEE 802.15.4 WPANs are used to monitor an urban area, by collecting data that has to be delivered, in a multi-hop fashion, to a specific node (called *PAN coordinator*) controlling the network. As a consequence, the ultimate goal of autonomic IEEE 802.15.4 networks is to create self-managing procedures to face the dynamism of scenarios where these networks will be used, i.e., they should:

- Be able to reconfigure after topological changes (nodes join and leave a WPAN)
- Present efficient topologies in terms of network resources use.

While the first goal is pursued by the IEEE 802.15.4 standard which defines MAC procedures to associate and de-associate nodes to a WPAN, the second one has not been deeply addressed. For this reason in [1] we focused on the impact of the PAN coordinator's position on the network performance, showing that the choice of the node acting as PAN coordinator highly affects energy consumption and delivery delay during traffic management. On the basis of this analysis, in [2] we presented a study on the selection/election of the PAN coordinator: we first presented a centralized mechanism to select the best node for PAN coordination, then we proposed a distributed procedure that aims at moving the PAN coordinator role to a target position, in order to achieve energy saving during and delay reduction during data delivery.

In this paper, we define a protocol framework to implement through fully standard-compliant solutions the distributed procedure for PAN coordinator election proposed in [2]. The innovative contributions are:

- The definition of a fully IEEE 802.15.4 standard-compliant protocol framework to implement the PAN coordinator election of [2]
- The analysis of the performance advantages that our distributed procedure achieves in terms of topological characteristics compared with the IEEE 802.15.4.

As for the latter point we stress that we can achieve a prolongation of the network lifetime, thanks to our topology reconfiguration, also by considering the energy spent to implement our standard-compliant protocol framework.

The paper is structured as follows: Section 10.2 recalls the main characteristics of the topology formation and the impact of PAN coordinator's position on energy consumption, as defined by the IEEE 802.15.4 standard. In Sect. 10.3 we describe our protocol framework for topology reconfiguration. Section 10.4 shows the performance analysis. The conclusions of the paper are provided in Sect. 10.5.

10.2 Network Set-Up in IEEE 802.15.4 and Impact of PAN Coordinator Position on Energy Consumption

An IEEE 802.15.4 WPAN consists of one PAN coordinator and a set of nodes [5]. A typical network topology defined in the standard and adopted as basis for ZigBee networking is the so called cluster-tree, where nodes associated to a single

PAN coordinator are arranged in a tree with parent–child hierarchical relationships [10]. In the rest of the paper, we refer to a tree-based topology. Nodes of an IEEE 802.15.4 network can be Full Function Devices (FFDs), which allow the association of other nodes to the network and for this reason are also called *coordinators*, and Reduced Function Devices (RFDs), which do not permit the association of other nodes. The PAN coordinator is always a FFD, intermediate nodes allowing data relay (i.e., routers) are FFDs too, whereas a RFD is always a leaf of the tree.

The standard defines a set of procedures implemented by the PAN coordinator to initiate a new WPAN and by other nodes to join an existing WPAN. The PAN coordinator starts by selecting a suitable channel. The procedure adopted by nodes to join a WPAN is named *association procedure*. The operations performed by a node to join a WPAN are: (1) the node searches for the available WPANs, (2) it selects a coordinator belonging to the available WPANs and (3) it starts a message exchange with the selected coordinator to associate with it. The discovery of available WPANs is performed by scanning *beacon frames* broadcast by coordinators. The time is divided into superframes, each bounded by beacon frames that are transmitted periodically and that allow nodes to synchronize. Beacon frames include a *payload field*, having maximum size equal to 52 *bytes* [5].

The association procedure results in a *parent−child* relationship between the two nodes. The whole set of these hierarchical relationships defines univocally a tree rooted at the PAN coordinator. After the network formation, each node in the tree has a data structure, called *Neighbor Table*, which includes a description of the children (number and type) and of the parent of that node.

The *tree depth L* is the maximum distance (in terms of number of hops) from the PAN coordinator: it is affected by the position of the PAN coordinator and it highly impacts the energy consumption during data delivery. In fact, the IEEE 802.15.4 WPANs typically employ hierarchical routing protocols, where data generated by nodes and directed to the PAN coordinator are routed upward to the root tree along the parent–child relationships. Therefore, the energy consumption due to data transmission is proportional to the number of hops between the source node and the PAN coordinator, since every node involved in the path spends energy to receive data from its children and to send the same data to its parent. By assuming that: (i) all nodes generate a data packet directed to the PAN coordinator; (ii) nodes do not perform data aggregation; (iii) collisions at MAC layer are negligible; (iv) E_{TX} and E_{RX} are, respectively, the energy spent by a node to transmit and to receive a data packet at 1-hop distance, the overall energy consumption of the network for one packet transmission by all nodes, can be computed according to the following Equation:

$$E_{\mathrm{tot}} = (E_{\mathrm{TX}} + E_{\mathrm{RX}}) \times \sum_{l=1}^{L} l \times x_l$$

where x_l is the number of nodes at level l within the tree.

As a result, the reduction of the tree depth L can lead to a reduction of the energy consumption for data transmission. Also the number of nodes at a given

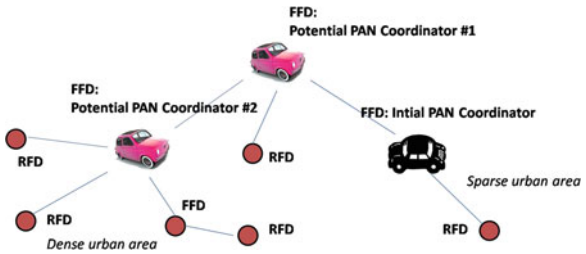


Fig. 10.1 Example of a WPAN scenario

level has a weight on the energy consumption. This simple model suggests that, if we are able to suitably select the position of the PAN coordinator, it is possible to control both the number of nodes at different levels of the tree and the tree depth. This motivates the proposed PAN coordinator election of [2]. For instance, Fig. 10.1 shows a WPAN that monitors an urban area, via FFD nodes on cars and FFD/RFD nodes on the roads: in this case, since each FFD can assume the role of PAN coordinator, it could be more indicated a PAN coordinator in dense urban area (left side of the figure) with respect to the initial one (right side), obtaining in this way a reduction of the tree depth. Although the initial PAN coordinator is a specific network node, the one that initiates the network set-up, in several scenarios this role can be changed during the network lifetime and assigned to different nodes [6, 8].

10.3 Framework for Topology Reconfiguration in IEEE 802.15.4

In this section, we describe the protocol framework for implementing in IEEE 802.15.4 networks our distributed procedure for PAN coordinator election. We remind that our distributed procedure runs on nodes already interconnected in a network topology formed according to IEEE 802.15.4 association rules.

10.3.1 Step 1: Initialization of Data Structures at FFDs

Our distributed protocol is based on the use of *beacon frames* sent by all FFDs in the network and, in particular, we use the payload field of these frames. The first two bits of this payload (called *XY*) are used as *flag bits* to discriminate the content of the rest of the beacon payload used by our protocol. Table 10.1 summarizes the meaning of these two bits, with reference to the *n*th iteration of our distributed procedure as will be better illustrated in the following.

Let us consider an IEEE 802.15.4 network with N_F FFDs and tree depth L. At the end of the association procedure, each FFD i (with $i = 1, 2, \ldots, N_F$) is able to compute a vector V_i whose jth element $V_i[j]$ indicates the number of nodes that are

Table 10.1 Values and meaning of the flag bits of *beacon frames*

Bit X	Bit Y	Content of the payload field
0	0	Empty space
0	1	Vector V_{i^n} of the generic FFD i^n
1	0	Vector V_{p^n} of the PAN coordinator p^n
1	1	Values of g_{i^n}, L_{i^n} and f_{i^n} computed by the generic child i^n of p^n

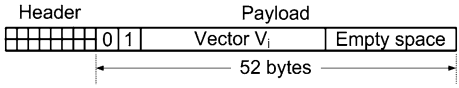


Fig. 10.2 Beacon frame including the vector V_i

j hops away from the node i in the subtree having i as root. Therefore, V_i is a description of the subtree having i as root node.

The computation of this vector starts from the FFDs which are leaves of the network tree (i.e., from the coordinators at level L): each FFD at level L sends its vector (which is empty, since it does not have any children) to its parent by using the payload field of the *beacon frames*, as shown in Fig. 10.2, with $XY = 01$. Nodes at level $L - 1$ are aware of the number of FFDs and RFDs that they have as child nodes, thanks to the *Neighbor Tables*, therefore they can build their vectors once received the beacon frames from all FFDs that they have as children. At this stage, nodes at level $L - 1$ can send their vectors towards their parents, by using the payload field of the *beacon frames* and this process keeps going, until the PAN coordinator is able to build its own vector V. Therefore, being L the tree depth, the initialization process ends after L superframes.

In Fig. 10.3 it is shown an example of computation of V_i performed by each FFD. After the IEEE 802.15.4 network formation each node has only a partial view of the topology: in fact, by using its *Neighbor Table*, each node is able to know only the number of its direct children (Fig. 10.3a). After the beacon frames exchange, each FFD can merge the information received by its FFD children and the one contained in its *Neighbor Table*, obtaining an updated description of the subtree having itself as root (Fig. 10.3b).

10.3.2 Step 2: Choice of the PAN Coordinator

The distributed procedure presented in [2]. At each iteration, the PAN coordinator role switches from the current PAN coordinator to one of its children (through the information exchange specified in the following), if there exists one child of the current PAN coordinator able to guarantee better network performance, once elected as new PAN coordinator. This role exchange is triggered only if the resulting network topology reshaped with the new PAN coordinator fulfils some topological properties (summarized in terms of three parameters, g, L and f,

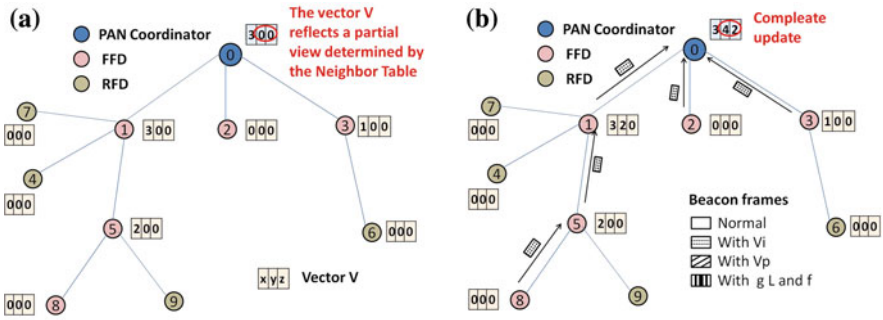


Fig. 10.3 Computation of vectors V_i. **a** Partial view of the topology by using only the Neighbor Table. **b** Topology update by using beacon frames exchange

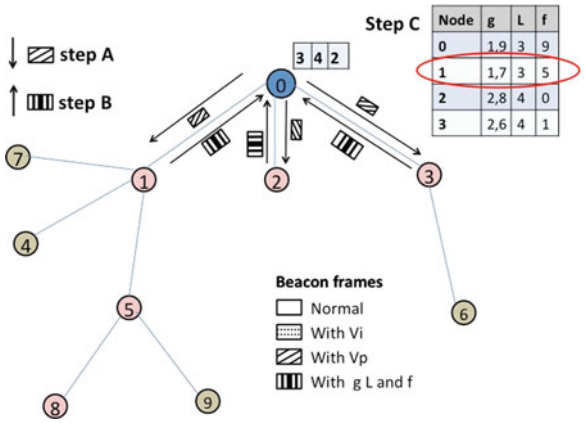


Fig. 10.4 Example of a generic iteration of PAN coordinator election

described in the following) that heavily affect network performance, otherwise the procedure ends.

Without loss of generality, let p^n the PAN coordinator at the beginning of iteration n (with $n = 1, 2, \ldots$) and V_{p^n} its own vector computed at *Step 1*. The PAN coordinator sends V_{p^n} towards its children, by using the payload field of its *beacon frame*, with $XY = 10$ (step A of the Fig. 10.4). Once received this vector from the PAN coordinator, each child i^n of p^n (with $i^n = k_{1^n}, k_{2^n}, hellip;, k_{m^n}$, where m^n is the number of FFDs children of p^n) is able to compute the values of three parameters [2]: the mean level of a node in the tree g_{i^n} and the tree depth L_{i^n} if i^n was elected as PAN coordinator, and the number of its descendants f_{i^n}, that is the number of nodes of the subtree having i^n as root. Then all the FFDs children of the current PAN coordinator send to p^n their values of these parameters, by using, also in this case, the payload field of their *beacon frames*, with $XY = 11$ (step B of the Fig. 10.4). After receiving the three parameters from all its FFD children, p^n computes the best (i.e., the minimum) values of g and L, taking into account also

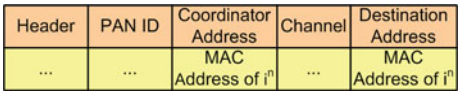


Fig. 10.5 Message sent by p^n to switch the role of PAN coordinator to i^n

its own values (step C of the Fig. 10.4). If there is a node i^n, among the FFDs children of p^n, that guarantees the minimum values of these parameters, it is elected as new PAN coordinator for the next iteration of the distributed procedure, which goes on, otherwise the procedure ends. The exchange of roles between p^n and i^n is achieved by using the *coordinator realignment command* defined in [5].

In particular, at the end of the nth iteration of our procedure, p^n sends to i^n the *coordinator realignment command*, specifying the MAC address of i^n into the *coordinator address field*, as shown in Fig. 10.5. When the node i^n receives this message, it assumes to be the new PAN coordinator, therefore it immediately runs a new iteration of this algorithm.

10.3.3 Step 3: Election of the Network's PAN Coordinator

At the end of the distributed procedure, i.e., when the PAN coordinator does not have any children able to achieve a more energy–efficient topology reconfiguration, it has to communicate to all nodes that it is definitively the PAN coordinator of the network. Therefore it sends, as in *Step 2*, a *coordinator realignment command*. The difference is that now the MAC address of the final PAN coordinator is inserted into the *coordinator address field* and the value $0 \times ffff$ into the *destination address field*, to trigger a broadcast propagation of this message into the network.

10.3.4 Topological Change Management

A significant requirement that should address our procedure is a prompt reaction to changes in the topology. In fact, during the network lifetime, it may happen that new nodes join the network or already associated nodes lose the network connection.

In such cases, every FFD that reveals a topological change immediately updates the vector V and sends it in its beacon frame with $XY = 01$. The information associated to the topological change quickly propagates in the network via the beacon frames, until it reaches the PAN coordinator. At this point, the PAN coordinator is aware of the topological change and can run our distributed procedure. On the whole, the idea to periodically transmit the vector V within each beacon frame, makes our protocol framework robust to topological changes and

Table 10.2 Simulation scenarios

Number of nodes, N	Side of the square area, a (m)	Transmission range, T_R (m)
20	25	15
50	62.5	20
100	125	25
150	187	27
200	250	30

allows to easily manage them, independently from the number and the rate at which they occur.

If topological changes happen during an execution of our distributed procedure, they are considered by the new PAN coordinator of the network just after the algorithm is completely ended, in order to avoid inconsistencies in the data structures.

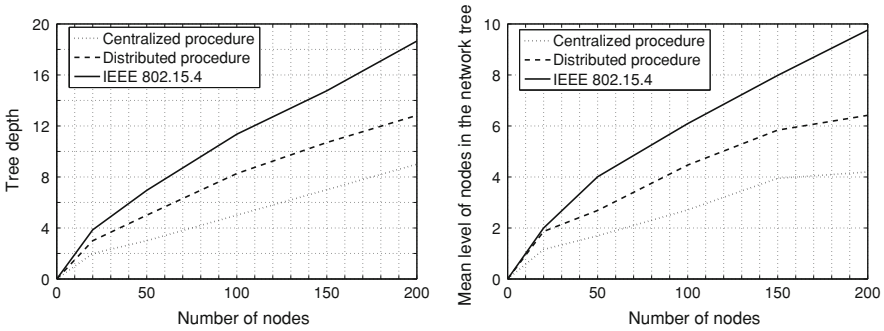
10.4 Performance Evaluation

The evaluation of our distributed procedure has been performed in two steps. In the first one, we compared the topologies resulting from our distributed procedure with the native IEEE 802.15.4 network trees and with topologies archived with a centralized procedure proposed in [2], in terms of resulting tree depth and mean level of nodes. In the second step, we compared the network lifetime of IEEE 802.15.4 trees and the same topologies reconfigured with the distributed procedure, under specific traffic conditions in the network.

In both cases, we considered a network scenario consisting of N nodes (with N_F FFDs), randomly deployed in a square area of side a, having transmission range T_R, according to Table 10.2. For each value of N, we simulated the IEEE 802.15.4 network set-up, by using *Network Simulator 2* (Ns-2 [9]) which implements the IEEE 802.15.4 standard. In all scenarios, the PAN coordinator at the beginning of the network formation process has been randomly chosen.

10.4.1 Comparison in Terms of Resulting Topology

Figure 10.6 shows a comparison in terms of tree depth and mean level of nodes in the network tree of the three aforementioned procedures with $N_F = \lceil 0.33 \times N \rceil$. As expected, the centralized procedure achieves the topology with the minimum tree depth and mean level of nodes, since as explained in [2] the parent–child relationships are optimized with a centralized algorithm. For this reason it can be considered as the upper bound of the topological performance even if it cannot be easily implemented in an autonomic IEEE 802.15.4



(a) Comparison in terms of tree depth.

(b) Comparison in terms of mean level of nodes in the resulting network tree.

Fig. 10.6 Comparison between the centralized and distributed procedures and the IEEE 802.15.4

network. On the other hand, the topology reconfiguration performed by the distributed procedure achieves advantages compared with the IEEE 802.15.4 topologies, since, after this reconfiguration, there is a reduction of tree depth and mean level of nodes.

10.4.2 Comparison in Terms of Energy Efficiency

The simulation scenarios described in Sect. 10.4 have been used also for a comparison in terms of energy consumption. After the network set-up performed with Ns-2, we reconfigured the network trees by applying our procedure on IEEE 802.15.4 topologies. Then, we simulated a *Constant Bit Rate* traffic, with the transmission of one packet by all nodes towards the PAN coordinator in each superframe, and we evaluated the network's energy consumption of IEEE 802.15.4 topologies and the same after the reconfiguration. We point out that for this evaluation, we also considered the energy spent by nodes to run our procedure. Table 10.3 summarizes the main simulation assumptions.

In Fig. 10.7 the percentage of network lifetime lengthening for the topologies reconfigured with our procedure compared with the IEEE 802.15.4 topologies is reported, for 2 different values of the number of FFDs. In all scenarios, after topology reconfiguration, there is a consistent increase of the network lifetime, which goes over 50% when $N = 200$ and $N_F = \lceil 0.50 \times N \rceil$. This means that our procedure achieves a topology reconfiguration that significantly reduces energy consumption. The advantage in terms of network lifetime increases when $N_F = \lceil 0.50 \times N \rceil$ (compared with the case when $N_F = \lceil 0.33 \times N \rceil$), since there are more nodes in the network able to be PAN coordinator, therefore our procedure achieves a more efficient reconfiguration.

Table 10.3 Simulation assumptions and parameters

Traffic type	Constant bit rate
Routing protocol	Hierarchical [3]
Size of data packets	25 bytes
Initial energy of nodes	1 J
Energy spent by anode to transmit/receive one data packet at $1 - hop$ distance	1 μJ/bit [7]
Energy spent by a node to run a single computational instruction	10^{-1} nJ/instruction [7]

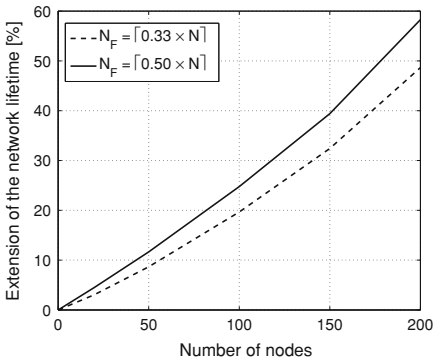


Fig. 10.7 Network lifetime lengthening after topology reconfiguration, for two different values of N_F

10.5 Conclusions

In this work we proposed a standard-compliant solution to implement in IEEE 802.15.4 a distributed procedure aiming at an energy–efficient topology reconfiguration, through the election of a suitable PAN coordinator. We presented a performance evaluation, which shows that topologies reconfigured according to our procedure achieves a significant reduction of energy consumption, compared with the same topologies after the native IEEE 802.15.4 network set-up.

Acknowledgments This work has been partially supported by the IT-funded FIRB/PNR IN.SY.EME. (protocol number: RBIP063BPH).

References

1. Abbagnale, A., Cipollone, E., Cuomo, F.: Constraining the network topology in IEEE 802.15.4. In: Annual Mediterranean Ad Hoc Networking Workshop, MED-HOC-NET '08 (2008)
2. Cipollone, E., Cuomo, F., Abbagnale, A.: Topology reconfiguration in IEEE 802.15.4 WPANs for emergency management. In: 8th IEEE International Conference on Pervasive Computing and Communications, PERCOM 2010, pp. 352–357 (2010)

3. Cuomo, F., Luna, S.D., Monaco, U., Melodia, T.: Routing in ZigBee: benefits from exploiting the IEEE 802.15.4 association tree. In: IEEE International Conference on Communications 2007, IEEE ICC '07, pp. 3271–3276 (2007)
4. Dobson, S., Denazis, S., Fernández, A., Ga, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A survey of autonomic communications. ACM Trans. Auton. Adapt. Syst. **1**(2), 223–259 (2006)
5. IEEE 802.15.4-2006 Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) (2006)
6. Jung, S., Chang, A., Gerla, M.: Comparisons of ZigBee personal area network (PAN) interconnection methods. In: 4th International Symposium on Wireless Communication Systems, ISWCS 2007, pp. 337–341 (2007)
7. Sentilla Corporation (ex Moteiv Corporation): http://www.sentilla.com
8. Sulaiman, T., Sivarajah, K., Al-Raweshidy, H.: Improved PNC selection criteria and process for IEEE 802.15.3. IEEE Commun. Mag. **45**(12), 102–109 (2007)
9. The Network Simulator, ns-2: http://nsnam.isi.edu/nsnam/index.php
10. ZigBee Specification: http://www.zigbee.org (2008)

Part III
Security in Online Social Networks

Chapter 11
On the Concept of Trust in Online Social Networks

Henric Johnson, Niklas Lavesson, Haifeng Zhao and Shyhtsun Felix Wu

Abstract Online Social Networks (OSNs), such as Facebook, Twitter, and Myspace, provide new and interesting ways to communicate, share, and meet on the Internet. On the one hand, these features have arguably made many of the OSNs quite popular among the general population but the growth of these networks has raised issues and concerns related to trust, privacy and security. On the other hand, some would argue that the true potential of OSNs has yet to be unleashed. The mainstream media have uncovered a rising number of potential and occurring problems, including: incomprehensible security settings, unlawful spreading of private or copyrighted information, the occurrence of threats and so on. We present a set of approaches designed to improve the trustworthiness of OSNs. Each approach is described and related to ongoing research projects and to views expressed about trust by surveyed OSN users. Finally, we present some interesting pointers to future work.

Keywords Social networks · Security · Trust

H. Johnson (✉) · N. Lavesson
Blekinge Institute of Technology, 371 39, Karlskrona, Sweden
e-mail: Henric.Johnson@bth.se

N. Lavesson
e-mail: Niklas.Lavesson@bth.se

H. Zhao · S. F. Wu
e-mail: hfzhao@ucdavis.edu

S. F. Wu
e-mail: wu@cs.ucdavis.edu

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_11,

11.1 Introduction

Today the Internet has become an important world wide network that connects a tremendous amount of groups and people. These users are further exploiting the network in a way that its creators probably never imagined, with streaming applications, e-commerce, cloud computing, mobile devices and Online Social Networks (OSNs). Therefore, the Internet is feeling the strain and is struggling to deal with the ever increasing demands placed on it. However, what is great about the Internet is that anyone with an address on the network can contact anyone else who has one. But that is also what is terrible about it. Global connectivity (IP addresses and e-mail) means you have no way to prevent large-scale attacks, citing as an example recent digital assaults that have temporarily shut down popular sites such as Twitter. At the same time you are getting convenience, you are actually giving people the power to do damage [1].

In recent years we have seen a dramatic increase and a growing popularity of OSNs. An OSN consists of a virtual social graph where users (nodes) are connected with each other through a relationship, which forms the edges of the social graph. OSNs services for an individual are: (1) to create a public or semi public profile where they share personal information such as name, contact, interests (2) to establish a social circle of friends for information sharing and communication (3) to view and traverse friends' profiles and private information (4) to carry out real time and non-real time communication with friends in the form of comments, private messaging, chatting, picture tagging etc, and (5) to use a lot of third party applications that range from gaming to advanced communication, virtual gifts, event management, and so on [2]. The Internet, keeps on the tradition of providing different communication and information sharing services. OSNs represent a recent type of communication and socializing platform [3], which is highly welcomed by the Internet users. Unlike the traditional web which revolves around information, documents, and web items, the concept of OSNs revolve around individuals, their connections and common interest-based communities. Examples of popular OSNs are Facebook, Twitter and MySpace.

Although OSNs provide a lot of functionalities to their users, their enormous growth has raised several issues such as scalability, manageability, controllability, and privacy. OSNs give rise to trust and security threats over the Internet, more severely than before. Trust has been on the research agenda in several disciplines such as computer science, psychology, philosophy and sociology. The research results show that trust is subjective and varies among people . We believe that this subjectivity of trust has been overlooked in the design of OSNs. However, it is a complex task to mimic real human communication and transfer the real world relationships into the digital world. With the entrance of OSN, the Internet is beginning to look like a virtual society that copy many of the common characteristics of the physical societies in terms of forming and utilizing relationships. This relationship is unfortunately in current OSNs assumed to be a symmetric and

binary relationship of equal value between the connected users and friends. In reality this assumption is wrong since a user has relationships of varying degrees.

The characteristics of trust can be defined as follows [4, 5]: *Trust is Asymmetric*: the trust level is not identical between two users. A may trust B, however, B may not necessarily trust A in the same way. *Trust can be transitive*: For instance, A and B know and trust each other very well, B has a friend named C in which A does not know. Since A knows B and trust B's friends, A might trust C to a certain extent. Then C has a friend named D whom either A or B knows. A could then find it hard to trust D due to the fact that, as the link between friends grow longer the trust level decreases. *Trust is context dependent*: Depending on the context one may trust each other differently, i.e., trust is context specific [6]. *Trust is personalized:* Trust is a subjective decision and two persons can have different opinions about the trustworthiness of the same person.

The value of OSN is to form genuine relationships with people who are either acquaintances or strangers in real life and to generate social informatics from people's interaction, which not only benefits the social network communicators or cooperators, but also helps in finding business merits. However, to magnify the value of OSN is not an easy task. It is important to understand how to create an architecture for the social network itself such that its value can be protected and how to leverage the value of OSN in communication with social informatics techniques. Therefore, the research objectives should focus on establishing a trustworthy OSN environment to handle cyber security and privacy issues. In the long run, the use of social informatics can hopefully influence the future Internet or system design.

11.2 Background

The Internet introduces a powerful way for people to communicate and share information. However, in terms of security and trust there exist some serious problems. One problem is the Internet's anonymity, in which the network has no social context for either the message or the sender. Compare that with ordinary life; people generally know the individuals they are communicating with, or have some sort of connection through a friend. If the network could somehow be made aware of such social links, it might provide a new and powerful defense against different cyber attacks and, perhaps more importantly, increase the level of trust within OSNs. If a more fine-grained friendship level scale is introduced, some issues will be resolved, since the user can then specify an appropriate level of friendship depending on how much information he or she wants to share. However, this level could then be selected quite subjectively and perhaps even arbitrarily by the users, thereby reducing the potential for resolving the aforementioned privacy and integrity issues. Moreover, it is difficult to establish a meaningful scale for the level of friendship since it has to correspond to a variety of subjective beliefs about trust, integrity, and privacy.

In OSN, different friendship intensities have different levels of default trust, which should be reflected in the privacy settings. The typical user does not change their privacy settings that often. Therefore, the default setting should be appropriate in order to actually preserve privacy. Since users normally don't do it themselves, it would be nice to have an automated way of finding out about the intensity of the friendship and derive the privacy settings from the intensity value. Users do not only change the privacy settings often enough, but also the OSNs have not provided a distinction between the types of friendships, as every relation is a friend, regardless of how intense the relation is. It is, therefore, a need to automatically identify the intensity of the friendship for the user. This way, the users' privacy settings could be reflective of the actual friendship intensity and automatically determined and set within the OSN for the specific user. This has to do with the relationship quality and an interesting approach to define the quality is to look at the interaction habits between users since it indicates differences in the strength of a relationship. This intuition is further discussed and supported in sociology research [7].

In social networks, individuals are connected with relations and these ties form the basis of their social network. These ties can in offline social networks (real life) be very diverse and complex depending on how close or intimate a subject perceives a relation to be. OSN, on the other hand, often reduces these connections to simplistic relations, in which you are friend or not [8]. These connections can be characterized by content, direction, and strength [9]. The intensity of a connection is also termed as the strength of that relationship. This characteristic indicates the closeness of two individuals or how powerfully two nodes are connected with each other in their social graph. Some users are prepared to indicate anyone as friends, while others stick to a more conservative plan. However most users tend to put other users on their list who they know or at least not actively dislike [8], i.e. this means that you can be friends in OSN even though the user does not even know or trust that person. This phenomenon is also very common in online social games, in which it is sometimes better to have as many friends as possible as a player. This could be a potential problem for social computing since OSNs might lose its value.

In offline social networks, the friendship intensity is a crucial factor for individuals while deciding the boundaries of their privacy. Moreover, this subjective feeling is quite efficiently utilized by human to decide various other privacy related aspects such as what to reveal and who to reveal. Therefore, in addition to other privacy and security threats, individuals can also face privacy threats from their own social network members due to the lack of trust and acquaintance. OSN users are unable to control these privacy vulnerabilities because: (1) Not enough privacy control settings are provided by OSNs, (2) The users do not know they have these settings, (3) The privacy controls are difficult to use, (4) Friendship is the only type of relationship provided by most OSNs to establish a connection between individuals, and (4) Individuals are unable to identify potential privacy leakage connections because their social networks consist of unreliable friends. Recently, some OSNs started to provide facilities to control information access but they are difficult to maneuver and normally overlooked by the users. Furthermore,

the relationship status between individuals tends to grow or deteriorate with the passage of time. Therefore, these privacy settings once set, may become meaningless after sometime. The binary nature of a relationship makes privacy very uncontrollable for OSN users. In these circumstances, the estimation of friendship intensity is quite useful to identify internal privacy threats.

For OSN, we have seen an increment use and development of social computing applications. Most, if not all, of these applications handle the issue of trust in a very ad hoc way. OSNs such as Facebook provide a naive trust model for users and application developers alike, e.g., by default all your Facebook friends have equal rights to accessing the information such as profile, status update, wall post and pictures related to a particular user. It is therefore of importance that application developers consider the trust and security issues within the scope of the application itself. Another interesting issue related to OSN is the ongoing data mining in which most OSN providers allow anybody to crawl their online data. The social computing paradigm has dramatically promoted the possibility to share social information online. Therefore, trust and security become increasingly challenging due to the speed and extent to which information is spread. On the other hand, how can this type of information sharing promote, e.g., the detection of malicious activity? The utilization of OSNs can further support authorities to handle crisis risk management and crisis management in a more efficient way. i.e., both by information dissemination and collection using OSNs.

While the popularity of OSN services, like Facebook and MySpace, are fast growing, some concerns related to OSN architecture design, such as privacy and usability, have emerged. For example:

- Friendships are not well differentiated. In reality, our friends can be classified with different circles, like families, colleagues, high school classmates, and etc. Furthermore, even in the same friend circle, we may stay closer to some people than the others. Even though current OSN services provide basic mechanism like friend list which provides more flexibility than before, it is still hard to differentiate the tie strength and the friendship quality and intensity between users.
- Personal information might be misused and privacy violation is also an issue of concern. Facebook provides various third-party applications that get access to personal information. Without surveillance and control on the applications they have joined, users personal information can be disclosed by malicious activity [10].
- All applications installed on Facebook use the same underlying social network, which is not only unnecessary, but also may result in privacy violation. Even if two users only want to cooperate or play together in just one application, they have to build friendship in Facebook. This may lead to unnecessary personal information disclosure.
- If a user posts something on his/her wall or someone else posts on the wall, all his/her friends granted with permission can see the post simultaneously. However, in real world, personal updates are spread asynchronously, probably from

intimates to general friends. A similar asynchronous information spreading mechanism is also needed for OSN.

11.3 State-of-the-Art

In general, trust can be defined as "The willingness of a party to be vulnerable to the actions of another part based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party" [11]. In real life (face to face), trust is a critical determinant of sharing information and developing new relationships [12, 13]. Therefore, trust should be an integrated concept in any network. There are in general two classes of trust systems [14]: a credit based system where the nodes are receiving credits and each message consumes a defined amount of credits [15]. The other part is reputation systems [16, 17]. The trustworthiness is then defined to be the probability that the next interaction is wanted. The reputation can be used on either a global scale with all other nodes or on a local scale for each neighbor.

An interesting project related to secured OSN is Safebook [18] that leverages the trust relationships that are part of the social network application itself. Safebook is a decentralized privacy preserving OSN that is governed by the aim of avoiding centralized control over the user data. The decentralization is provided by the use of peer-to-peer technology. The work presented in [19] describes a new application of threshold-based secret sharing in a distributed OSN. The developed mechanism will select the most reliable delegates based on an effective trust measure. Relationships between the involved friends are used to estimate the trustworthiness of a delegate.

A number of improved encryption solutions have been presented in the literature that are based on improved encryption and overall privacy management systems for OSN websites [20, 21]. Moreover, in a study conducted by Xi et al. [22], two different forms of private information leaks in social networks are discussed and several protection methods are reviewed. However, most of the current privacy improvement solutions add substantial amounts of user interface complexity or violate social manner. A good interface should not restrict or block users from contributing, sharing or expressing. Thus, a privacy preserving method within social norms is a difficult yet important research aim.

Gilbert and Karahalios [23] reflect upon the fact that social media treats all users the same: trusted friend or total stranger. In reality, Gilbert and Karahalios argue, relationships fall everywhere along this spectrum and in social science this topic has been investigated for a long time using the concept of tie strength. A quantitative experiment conducted in [23] shows that a predictive model that maps social media data to tie strength using a dataset of over 2,000 social media ties manages to distinguish between strong and weak ties with over 85% accuracy.

From a psychological or developmental point of view, a large amount of work has been conducted on establishing friendship measures, both for the real world

and for the social media. For example, Punamaki et al. [24] study the relationship between information and communication technology and peer and parent relations while Selfhout et al. [25] focus on differentiating between the perceived, actual, and peer-rated similarity in personality, communication, and friendship intensity when two people get acquainted. Whereas Steinfield et al. [26] and Vaculik and Hudecek [27] investigate aspects, such as the building of self-esteem and the development of close relationships in a social network setting, Rybak et al. [28] try to establish a measure for friendship intensity in a real world setting. We believe that the OSN friendship concept could, and should, be more thoroughly investigated in order to improve the privacy and integrity of OSN users. Additionally, we argue that refined friendship level indicators, along with the content and interaction analysis required to develop these indicators, would enable the improvement of trust in OSNs.

As far as calculation of friendship intensity is concerned, an interesting study conducted by Banks et al is presented in [29]. They have introduced an interaction count method. In this method, they suggested to count selected interaction types between individuals in order to calculate the friendship strength. In addition to provide a novel intensity calculation method, they also suggested a framework that utilizes calculated friendship intensity for better privacy control in OSNs. In [30] the authors also utilized the number of interactions between individuals for the improvement of a recommendation process in OSNs.

To improve the current infrastructure defects of OSN, the Davis Social Link (DSL) research group[1] has developed an infrastructure: FAITH (Facebook Applications: Identifications Transformation & Hypervisor) to provide more trustworthy and flexible services to users. FAITH itself is a Facebook application, but works as a proxy between users and Facebook. It can hook other applications and provide services to the applications. Figure 11.1 describes the architecture of FAITH, in which FAITH provides three major functions:

1. Each application hooked under FAITH is monitored by FAITH. All Facebook Application Programming Interfaces (APIs) that the application called are logged and available for the user to review. The log information helps the users to keep track of their personal information executed by an application and also for the system to perform anomaly/intrusion detection.
2. Users can customize the API related to their personal information that an application can call. If a user feels an API is not necessary for an application, he/she can block the API so that the application can not access the user's relevant information. API customization prevents applications from maliciously impersonating the user.
3. Virtual social networks are generated and maintained for each application. FAITH initializes a friendship network for each application from Facebook. Users can then add or remove their friendships in an application hooked under Faith without affecting their friendships in Facebook. Similarly, they can also

[1] http://dsl.cs.ucdavis.edu/

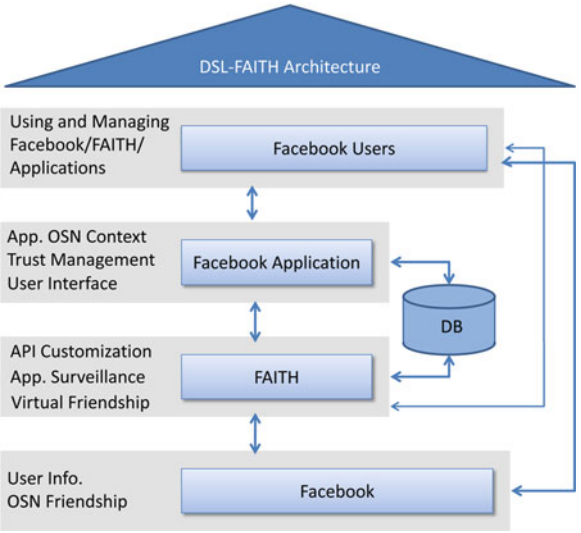


Fig. 11.1 The architecture of FAITH

disconnect the friendships with others just within an application but not affect their friendships in Facebook. Virtual social networks provide users with flexibility to find new friendships and differentiate friend circles while protecting their privacy.

The DSL research group has developed several applications in FAITH to improve OSN user experience, boost communication and cooperation with social informatics techniques. There are two developed applications that we would like to highlight:

- SoEmail [31]: SoEmail stands for Social E-mail. Traditional email systems provide too little control to the recipient, so the recipient can not prevent from receiving spam. SoEmail incorporates social context to messages using an OSN's underlying social graph. If a sender wants to email a receiver, he/she should find a path on the social graph leading to the receiver. It mimics the real work social network. If two people are not friends and one wants to know the other, he/she need to find intermediate people with his/her social relationships to recommend him/her. If the receiver dislikes the email content, he/she can punish the social path from the sender, which decreases the trust value [32] from him/her to the sender.
- SocialWiki [33]: SocialWiki stands for social wiki systems. In current large wiki systems, a huge amount of administrative efforts are required to produce and maintain high quality pages with existing naive access control policies. SocialWiki leverages the power of social networks to automatically manage reputation and trust for wiki users based on the content they contribute and the

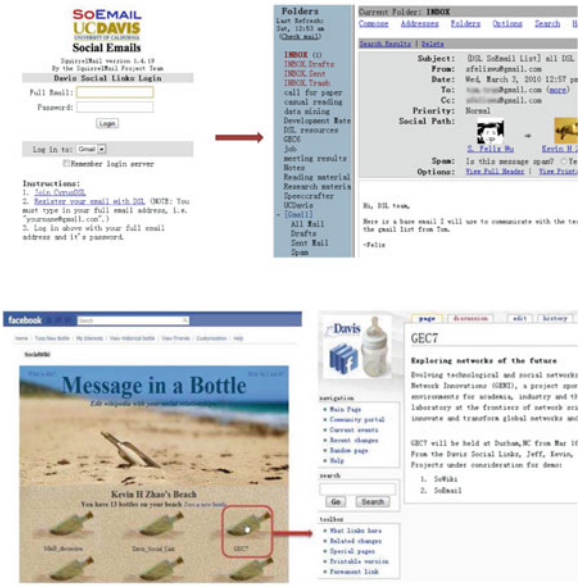


Fig. 11.2 SoEmail & SocialWiki

ratings they receive. Although a SocialWiki page is visible to everyone, it can only be edited by a group of users who share similar interests and have a certain level of trust with each other. The editing privilege is then circulated among these users in an intelligent way to prevent spam. For a snapshot of SoEmail and SocialWiki see Fig. 11.2.

11.4 Security Perception and Use of OSNs

It is quite evident that today's OSNs, such as Twitter, Facebook, Myspace, and Spotify, together represent a useful and exciting portfolio of ways to communicate and share information whether it be music, videos, news, or facts and fiction about everyday life. Intuitively, the different OSN services emulate different parts of our lives and of the traditional ways we communicate and share information.

However, as can be observed in both scholarly literature and mainstream media, important concepts such as privacy, integrity, and trust, need to be redefined or at least considered using different approaches in OSNs compared to what is the norm in traditional social networks. Strong criticism has been raised against how privacy, trust, and security are implemented in the aforementioned OSNs. Perhaps most notably, there are countless cases in which the general population have published personal information in an OSN without really considering the consequences of making such content available on the Internet. For example, last year,

the wife of the then new head of the British MI6 managed to cause a security breach and left his family exposed after publishing photographs and personal details on Facebook.[2] In the aftermath of this and other similar events, the privacy concerns and online security awareness of OSN users have been frequently discussed.

Recently, an anonymous survey regarding OSN usage, addressing the problems of online privacy and trust, was conducted at Blekinge Institute of Technology [34]. The group of survey respondents consisted of 212 individuals from 20 nationalities. Out of the 212 respondents, 86% were male ($n = 182$) and 14% were female ($n = 30$). The skewed male to female ratio is primarily due to the low number of female students at the School of Computing at which the survey was conducted. The participants were furthermore divided into three age groups; younger than 20, between 20 and 40, and older than 40. Quite intuitively, 96% of the participants belonged to the middle group (aged 20–40).

The aim of the survey was to gather information about the perception and use of OSNs, especially related to trust, privacy concerns, and integrity issues. In this particular survey, the scope was limited to Facebook users for several reasons: Facebook is one of largest and most well-known OSNs and, additionally, it has arguably been criticized heavily in both the mainstream media and in scientific work on the subject of lacking or too complex security and privacy settings.

A web-based survey questionnaire was created, which featured 21 closed questions. The questionnaire is logically divided into two parts: part one covers privacy-related aspects of Facebook usage and part two addresses the OSN interaction habits between the respondent and those users he or she considers as good friends. We will get back to the concept of good friends and its applicability to the enhancement of trust and privacy later.

As a basis for performing the analysis of the survey questions related to privacy and trust, the questionnaire featured a number of questions regarding the level of Internet experience as well as the frequency of Facebook usage. The results show that a majority of the respondents are active to very active Facebook users and most respondents regard themselves as being either expert or above average level Internet users. Thus, the survey results should be interpreted with this experience level of the respondents in mind. In other words, the results are hardly generalizable to the general population but should give an indication of what the thoughts and motivations of a large group of OSN users on the concepts of privacy and trust.

It is evident from the results of the survey that most users are connected to between 100 and 200 friends in the OSN. Almost 80% of the respondents have more than 50 friends in their network. A minority of the respondents have more than 1,000 friends in the OSN network. Naturally, if respondents were asked to estimate how many friends they have in their real world social network, it is plausible to assume that the number of friends would generally be much lower than

[2] The Times, http://www.timesonline.co.uk/tol/news/uk/article6639521.ece

what is observed in the OSN survey. However, in the real world social network, friends can usually be organized into groups of different levels of friendship (e.g., in terms of how long ago the friendship started or how active friends are in meeting and communicating with each other).

It is interesting to note that approximately 77% of the respondents are concerned about the privacy issues of Facebook and about 70% are actively avoiding to publish private data on Facebook due to privacy concerns. Judging by the frequency of alarms about OSN privacy breaches and issues, these figures either do not reflect the views and actions of the general population or it can be suspected that the media blows OSN privacy issue stories out of proportion for some reason.

A majority of the respondents (58%) are of the opinion that Facebook third party applications represent the biggest threat to privacy. The remaining respondents believe that their friends (16%) and friends of their friends (26%) pose a greater threat to privacy. A quite staggering 66% of individuals suspect that at least one friend in their online network could have a malicious intent towards their privacy. Its also curious to observe that, 28% of those respondents who stated that their online friends could represent a privacy threat, report that they still add unknown people (e.g., people they do not know in the real-world) to their network. The willingness of individuals to expose private data to the friends they have in an OSN is arguably a factor that could reflect the confidence level the individuals attribute their friend network in the OSN. Close to 90% respondents only want to share their private data with selected friends in their network. In terms of general opinions regarding the security of the OSN, 56% of the respondents state that the Facebook privacy settings are too difficult to use. One interpretation of this results is that these users of Facebook are unsure whether the security settings they select really reflect their personal opinion on privacy and integrity.

11.5 User Interactions and Their Implications

As have been previously discussed, the largest OSNs today, e.g., Facebook, employ a binary type of Friendship Intensity. That is, either two individuals have an established link as friends, e.g., a social link, or no direct link between them exists. From a privacy and integrity point of view, the binary friendship type poses several potential problems. For example, since users can only reject or accept a friend request (as no intermediate levels of friendship exist), there is a risk of unintentional sharing of personal information. The level of sharing is dictated by the OSN privacy settings, which many users neglect to get informed about or find it too time consuming and cognitively demanding to manually adjust. From the point of view of trust, we constantly assign different levels of trust toward our friends, relatives, business contacts, and so forth. The OSN binary level of friendship is not sufficient as a means to implement this real world concept of trust.

An important and open research question is whether (and how) the level of trust within OSNs can be increased by the use of social link content and friendship

intensity determination. The average user seems to find it cumbersome to manually adjust privacy and security settings in the OSN. It would probably be even more difficult for users to manually establish the intensity of their relationship to other users. Thus, formulated more specifically, the question is how to reliably determine friendship intensity by automatic analysis of OSN user information and interaction data and how this can be used to better identify and determine social groups.

It is evident that technological advances have resulted in a general change of lifestyle and expanded the focus of the global economy from production of physical goods to manipulation of information [35]. As a consequence, we rely more and more on storing and retrieving information in/from databases. The number and size of the databases grow swiftly. It is even argued that stored data is doubling every nine months. It is therefore becoming increasingly hard to extract useful information. It can be noted that data mining technologies have been shown to perform well at this task in a wide variety of science, business, and technology areas.

Data mining, or knowledge discovery, draws on work conducted in a variety of areas such as: machine learning, statistics, high performance computing, and artificial intelligence. The main problem studied is how to find useful information in large quantities, or otherwise complex types, of data. Although the nature of this problem can be very different across applications, one of the most common tasks is that of identifying structural patterns in data that can then be used to categorize the data into a distinct set of categories [35]. If these patterns can actually distinguish between different categories of data this implies that they have captured some generalized characteristics of each category. As it turns out, the area of machine learning provides a number of approaches to automatically learn this kind of concepts by generalizing from categorized data [35]. Moreover, regression problems, which have previously been studied in the statistics area of research, have also been revised and automated within the machine learning field. The question is whether data mining technologies, powered by machine learning theory, can be employed as a means to migrate the real world concept of trust to the OSN community.

A plausible approach for automatically determining friendship intensity would be that of establishing an appropriate friendship intensity model, based on Interaction Intensity [29], but with additional friendship aspects and by drawing on relevant work from both social science and information systems, e.g., [23–28]. Such a friendship intensity model could be used as a basis for determining which user and interaction data to extract from the OSN, and for learning how to organize and preprocess these data, so that data mining algorithms can be applied to automatically predict friendship intensity. Thus, based on the friendship intensity model and the knowledge discovered through data mining, it would be possible to establish a new friendship intensity measure. This measure would of course have to be empirically compared to the state of the art using experimental evaluations on simulated and real-world data. A practical issue related to obtaining real-world data is that most OSNs (for example, Facebook) restricts the number of access

attempts of user designed OSN applications. Thus, a quite elaborate batch-processing algorithm needs to be designed with deliberate restrictions concerning the number of accesses per some suitable time unit. Additionally, OSN friendship intensity may be defined as either a discrete or continuous metric, and the choice of metric type implies that either classification or regression learning algorithms are to be used for building the prediction model using the gathered data [35].

The data collection is believed to encompass a variety of data types such as text (e.g., personal descriptions, interests, observations, activities, and messages), numbers (e.g., years, months, and statistics like the number of sent and received messages), and categories (e.g., religious belief/political standpoint, events, locations). Thus, the learning algorithms selected for inclusion are required to handle these data types [35]. In the case where natural language is to be analyzed, the text needs to be transformed to an appropriate representation, such as the Bag-of-words model, which has been proven to work well for many text classification tasks, cf. [35].

11.6 Conclusions and Future Work

Although OSNs provide new and interesting functionalities that are very popular, the growth of these networks has raised issues related to trust and security. Examples of security concerns are that friendships are not well differentiated and personal information is unnecessarily disclosed and might be misused. If these issues are not prioritized the value of OSN might decrease. Therefore, the aim of our research is to establish a trustworthy OSN environment, in which the DSL research group has developed several novel applications that provide OSN users with flexible and secure services.

The results from a web-based survey questionary was presented that addressed the problem of online privacy and trust for OSNs. A majority (77%) of the respondent are concerned about the privacy issues and also of the opinion that third party applications are the biggest concern related to privacy. Most of the respondents also stated that the privacy settings on an OSN are normally too difficult to use.

The level of trust within a social network can in our opinion be increased by determine the friendship intensity. This is further discussed by the use of data mining to identify structural patterns in the interaction and non-interaction based data. One possible approach would be to establish an intensity model to determine different levels of friendship between your friends. In the future, we plan to develop a data mining framework and will utilize various classification and numerical predictation algorithms. Later on, we will validate the performance of this model on Facebook.

As part of future work, the DSL research group is continuously developing both FAITH and other applications. One feature in FAITH that is under development is Asynchronous Information Dissemination (AID). AID will be designed to increase

users' flexibility to control the updates they want their friends to view. With AID, each user will be able to assign a rule to a message before publishing. The rule defines who will see the message, at what time they can see the message and what operations (e.g., like/comment/share) they can do with the message. AID allows users to define rules of how to publish updates on their friends' walls asynchronously. It brings multiple benefits to OSN users. First, it can provide a fair OSN game environment to cope with controversial browser plug-ins (e.g., Snag Bar of Gamers Unite[3]). AID can also be applied to protect personal privacy if users do not want to disseminate personal news immediately. By employing semantic methods to extract message topics, we will further be able to automatically predict users' preference based on historical events. Besides providing flexibility to OSN users, AID also reduces network traffic and server load by delaying updates, canceling invalid or removed updates and preventing plug-ins from continuously scanning walls. In the near future, we further see a need for more applications leveraging social informatics in order to create and maintain a trustworthy Internet.

References

1. Gammon, K.: Networking: four ways to reinvent the Internet. Nature **463**(7281), 602–604 (2010)
2. Cutillo, L.A., et al.: Safebook: a privacy-preserving online social network leveraging on real-life trust. IEEE Commun. Mag. **47**, 94–101 (2009)
3. Boyd, D.: Social network sites: definition, history, and scholarship. J. Comput.-Mediat. Commun. **13**, 210–230 (2007)
4. Golbeck, J.A.: Computing and applying trust in web-based social networks. Ph.D. thesis, University of Maryland (2005)
5. Dey, A.: Understanding and using context. Pers. Ubiquitous Comput. **5**(1), 4–7 (2007)
6. Gray, E.L.: A Trust-based management system. Ph.D. thesis, Department of Computer Science and Statistics, Trinity College, Dublin (2006)
7. Granovetter, M.S.: The strength if weak ties. Am. J. Sociol. **78**, 1360–1380 (1973)
8. Boyd, D.: Friendster and publicity articulated social networking. In: Conference on Human Factors and Computing Systems (CHI 2004), Vienna, Austria 24–29 April 2004
9. Garton, L., et al.: Studying online social networks. Haythornthwaite, et al., Eds., ed: Journal of Computer-Mediated Communication, June (1997)
10. Skinner, C.: Phishers target facebook. PCWorld (March 01, 2009)
11. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. Acad. Manag. Rev. **20**(3), 709–734 (1995)
12. Lewis, J.D., Weigert, A.: Trust is a social reality. Soc. Forces **63**(4), 967–985 (1985)
13. Fukuyama, F.: Trust: The Social Virtues and the Creation of Prosperity. Simon and Schuster Inc, New York (1995)
14. Spear, M., Lu, X., Wu, S.F.: Davis social links or: how i learned to stop worrying and love the net. In: IEEE International Conference on Computational Science and Engineering, pp. 594–601 (2009)
15. Cohen, B.: Incentives build robustness in bittorrent. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.1911 (2003). Accessed 2003

[3] http://gamersunite.coolchaser.com/

16. Commerce, B.E., Jsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15 the Bled Electronic Commerce Conference (2002)
17. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation managament in p2p networks. In: WWW'03: Proceedings of the 12th international conference on World Wide Web. New York, NY, USA: ACM, pp. 640–651 (2003)
18. Cutillo, L.A., Molva, R., Strufe, T.: Privacy preserving social networking through decentralization. In: Wireless On Demand Network Systems and Services (2009)
19. Vu, L.H., Aberer, K., Buchegger, S., Datta, A.: Enabling secure secret sharing in distributed online social networks. In: Proceedings of the 2009 Annual Computer Security Applications Conference ACSAC '09, IEEE Computer Society, pp. 419–428. Washington D.C. (2009)
20. Baatarjav, E.A., Dantu, R., Tang, Y., Cangussu, J.: BBN-based privacy management system for Facebook. In: Proceedings of the 2009 IEEE International Conference on Intelligence and Security Informatics, pp. 194–196 (2009)
21. Baatarjav, E.A., Dantu, R., Phithakkitnukoon, S.: Privacy management for Facebook. In: Proceedings of the Fourth International Conference on Information Systems Security, pp. 273–286 (2008)
22. Xi, C., Shuo, S.: A literature review of privacy research on social network sites. In: Proceedings of the 2009 International Conference on Multimedia Information Networking and Security, pp. 93–97 (2009)
23. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proceedings of the 27th international conference on Human factors in computing systems, pp. 211–220. ACM press, New York (2009)
24. Punamaki, R.L., Wallenius, M., Holtto, H., Nygard, C.-H., Rimpela, A.: The associations between information and communication technology (ICT) and peer and parent relations in early adolescence. Int. J. Behav. Dev. **33**(6), 556–564 (2009)
25. Selfhout, M., Denissen, J., Branje, S., Meeus, W.: In the eye of the beholder: perceived, actual, and peer-rated similarity in personality, communication, and friendship intensity during the acquaintanceship process. J. Pers. Soc. Psychol. **96**(6), 1152–1165 (2009)
26. Steinfield, C., Ellison, N.B., Lampe, C.: Social capital, self-esteem, and use of online social network sites: a longitudinal analysis. J. Appl. Dev. Psychol. **29**(6), 434–445 (2008)
27. Vaculik, M., Hudecek, T.: Development of close relationships in the internet environment. Ceskoslovenska Psychologie **49**(2), 157–174 (2005)
28. Rybak, A., McAndrew, F.T.: How do we decide whom our friends are? Defining levels of friendship in Poland and the United States. J. Soc. Psychol. **146**(2), 147–163 (2006)
29. Banks, L., Wu, S.F.: All friends are not created equal: an interaction intensity based approach to privacy in online social networks. In: Proceedings of the 2009 International Conference on Computational Science and Engineering (CSE), pp. 970–974. Piscataway, NJ, USA, 29–31 August 2009
30. Katarzyna, M.: Recommendation system for online social network. Blekinge Institute of Technology, Master's Thesis in Software Engineering, Thesis no: MSE-2006:11, July (2006)
31. Tran, T., Rowe, J., Wu, S.F.: Social email: a framework and application for more socially-aware communications. In: SocInfo '10: Proceed- ings of the 2nd IEEE International Conference on Social Informatics, Austria (2010)
32. Tran, T., et al.: Design and implementation of davis social links OSN Kernel. In: 4th International Conference on Wireless Algorithms, Systems, and Applications, WASA 2009, pp. 527–540, Boston, MA, United states (2009)
33. Zhao, H., Ye, S., Bhattacharyya, P., Rowe, J., Gribble, K., Wu, S.F.: Socialwiki: Bring order to wiki systems with social context. In: SocInfo '10: Proceedings of the 2nd IEEE International Conference on Social Informatics (2010)
34. Ahmad, W., Riaz, A., Johnson, H., Lavesson, N.: Predicting friendship levels in online social networks. In: Proceeding of the 21st Tyrrhenian Workshop on Digital Communications: Trustworthy Internet, Island of Ponza, Italy (2010)
35. Lavesson, N.: On the metric-based approach to supervised concept learning. Ph.D. thesis, No. 2008:14, Blekinge Institute of Technology, Ronneby, Sweden, 2008 (2010)

Chapter 12
Participatory Sensing: The Tension Between Social Translucence and Privacy

Ioannis Krontiris and Nicolas Maisonneuve

Abstract Participatory sensing is a new research area that emerged from the need to complement our previous efforts in wireless sensor networks. It takes advantage of the emergence of rich-sensor mobile phones and their wide adoption, in order to turn people to producers of sensed data and enable new classes of collective applications. Unavoidably, this raises a lot of privacy concerns, as people are becoming sensors and give out a lot of personal information, like their location. If we choose to protect their privacy by anonymizing the data and completely hiding any identifying information, then the visibility of their contributions to others is lost. However, it is important to maintain this property, in order to support accountability on one hand and allow people gain reputation for their efforts on the other hand. In this book chapter we investigate which of the available technical solutions we need, in order to resolve this conflict and what are the research directions that emerge.

Keywords Privacy · Trust · Anonymity · Social networking

12.1 Introduction

Over the several past years, there has been a great amount of research on wireless sensor networks, using dedicated embedded devices for data collection, e.g., from the environment or an infrastructure. The deployments of sensor networks have

I. Krontiris (✉)
Chair of Mobile Business and Multilateral Security, Goethe University Frankfurt, Frankfurt, Germany
e-mail: ioannis.krontiris@m-chair.net

N. Maisonneuve
e-mail: n.maisonneuve@gmail.com

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_12,

been treated as peripheral networks attached to the mainstream domain of the Internet through a gateway, delivering in this way data to the end-users. However, the initial vision of connecting thousands of sensors that have been randomly disseminated into the environment ("smart dust") seems to be still far out of reach.

At the same time, the wide adoption of mobile phones in combination with the spread of the Web 2.0 paradigm on the Web recently created the right conditions for a new scope of research, often referred to as *participatory sensing* [1], which comes to complement our previous efforts in wireless sensor networks. Thanks to sensor-rich devices, geo-localised user-generated content can now be created any time, anywhere. Other sensors, besides geo-location chips, such as camera, gyroscope, light sensor or accelerometer started becoming more and more prevalent in mobile devices carried by billions of people, enabling new large-scale practices. So, the vision of a sensor data-sharing infrastructure emerged, where people and their mobile phone devices could provide sensor data streams in accessible ways to third parties interested in integrating and remixing data, enabling new citizen science campaigns [2, 3] and empowering local communities to manage their commons.

Participatory sensing regards end-users not only as consumers, but also as active producers of data and uses the sensors attached to the user as natural source of information.This new direction changes several underlying assumptions of typical sensor networks, as for example urban deployment, no pre-defined sink nodes, more powerful rechargeable nodes, mobility (humans, cars, etc.) and a new variety of sensors, attached closer to human beings and the context around them. With participatory sensing, the research area of sensor networks is moving into a direction, in which sensing networks will be several orders of magnitudes larger than the average existing classical sensor networks and where the short-term real-world impact may be much higher.

In the wireless network research community, participatory sensing is sometimes referred to as people-centric sensing [4], urban sensing [5] or mobile sensing [6]. But while all these terms are close, they emphasize different aspects. *People-centric sensing* focuses on the nature of the data collected, e.g. health data, food consumption or personal noise exposure, and it does not necessary refer to location-related data. *Urban sensing* emphasizes on the environment where the sensing process takes place, i.e., the urban space. *Mobile sensing* emphasizes on the mobility and the nature of the sensor device, i.e. the mobile phone. Finally, *participatory sensing* emphasizes on the participatory nature of some projects, which is the focus of this article.

Indeed, a lot of participatory sensing-related projects target only the individual level. An increasing number of mobile applications deliver self-monitoring services, for instance sensing their daily exposure to pollution, keeping track of their exercise activities, dietary habits, etc. While some of these services, like CenceMe [7], allow to share such information within social networks, they are mainly individual centric. However, in this chapter we focus on sensing projects that also target the community level and in which users sharing commons do not necessary get an individual and direct benefit from offering their sensing capabilities. They are rather motivated by a common cause or interest, similar to the participative paradigm of Web 2.0. Both

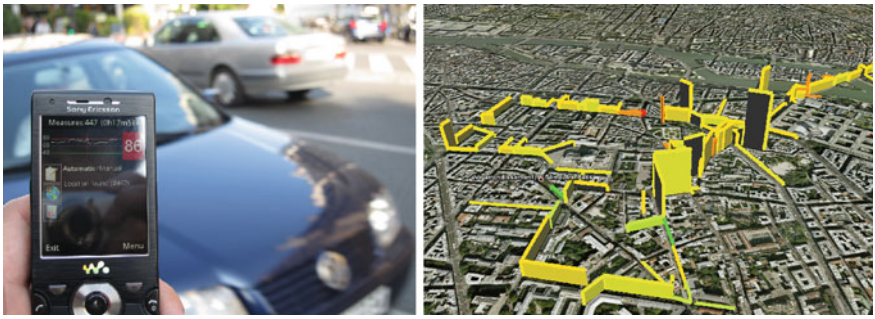


Fig. 12.1 The NoiseTube project [8]. A sensing project mixing mobile sensing applications with participatory and community building aspects to create a collective exposure map of noise pollution

levels are compatible. For example the NoiseTube [8] project enables citizens to measure their daily exposure to noise in their environment on one hand and report such information to produce a noise map representing the collective exposure experienced by the local people, on the other hand (see Fig. 12.1).

12.1.1 Problem Statement and Chapter Organization

In the first part of this chapter, we focus on participatory and accountability-related aspects. Currently, many projects call people to participate with the goal to collect sensor data. But these calls have been only moderately successful. The focus of research should be extended to investigate how we can make people involve more actively in participatory sensing projects. Secondly, like any participatory system, such as Wikipedia, participatory sensing is vulnerable to gaming and spamming. A major challenge is thus enabling broad user participation by making the system accountable for all data.

To tackle these two issues in the general context of online communities, Erickson and Kellogg [9] proposed to integrate *Social Translucence* features. Social Translucence is a term to refer to "digital systems that support coherent behavior by making participants and their activities visible to one another". The goal of such feature is to faciliate participation, self-organisation and *accountability*. Such social feature, present in current communities platforms (e.g. Facebook), could provide many benefits to the design of future participatory sensing projects.

At the same time, several research work is focusing on privacy issues of participatory sensing (see [10] for a survey). Indeed, with the mobile devices gathering sensor data from user's immediate environment and deriving user context, privacy concerns are rightfully raised. In this chapter we focus only on location-related data as they are commonly sensitive. So the goal is to prevent access to location information at all costs, making it tamper-proof against both (i) malicious

hackers with the desire to intrude on other people's privacy, and (ii) against companies profiling and accumulating users' location information for profit maximization.

Then the question that we raise in this chapter is: how can we maintain social translucence features to preserve participation and accountability, while preserving privacy at the same time? Indeed, protecting privacy will unavoidably limit the visibility and accountability of user contributions to the minimum.

In the second part of the chapter, to answer this question we explore the use of an anonymity-based approach: all data sent to the service provider do not include any identifying information of the sender. In the context of a social translucence design, we then investigate two related problems:

- In Sect. 12.4.2, we discuss about accountability. How to revoke the access credentials of users, who covered behind their anonymity, misbehave against the system?
- In Sect. 12.4.3, we discuss about maintening reputation and social recognition. How to enable users to accumulate reputation points and receive recognition for their contributions, even though these contributions were made anonymously?

12.2 Social Factors for Participation and Accountability

As in many community-based services, a key factor in participatory sensing lies in the leverage of participation in data gathering. Even though the ubiquity of mobile phones makes mass participation feasible, as attempted in [11], it remains questionable how the general public can be motivated to voluntary participate. In most cases, participatory sensing projects call users to volunteer and offer the sensing capabilities of their mobile devices without getting any immediate social benefits. What benefits would they gain that might compensate them for their efforts?

Furthermore like any participatory system, such as Wikipedia, participatory sensing is vulnerable to gaming and spamming. A major challenge is thus enabling broad user participation by making the system accountable for all data.

Well known Web 2.0 success stories such as Flickr, YouTube or Wikipedia prove that it is possible to actively involve people in community projects with no self-reflective benefits. The question is then, can we transfer online social practices from the digital world to the real world via mobile technology?

12.3 Social Translucence Design for Participatory Sensing

In the context of online communities, Kollock outlines three motivations that do not just rely on altruistic behavior on the part of the contributor: anticipated reciprocity, increased recognition and sense of efficacy [12]. Indeed, as pointed out

in [13], the Web 2.0 phenomenon contradicts many predictions regarding the form of cooperation and community building that were encouraged by the founders of the Web. As shown in studies of bloggers or Wikipedia [14], the motivations of contributors do not fit into a single category. They are not either utilitarian, targeting to maximise personal interest or just altruistic, motivated by a desire to volunteer and be part of a community. Users generally first have individualistic motivations when they begin to make visible personal production (e.g. blog posts). Such tendency to get social recognition and attract attention by making their contributions public appears to develop a greater number of interpersonal relations than expected, although the links between individuals are weak. From such dense interaction emerge opportunities of cooperation, transforming user goals from individual interest to more collective.

A *social translucence* design has been proposed by Erickson and Kellogg [9] to make participant's contributions and activity visible to the community. Social Translucence is a term to refer to "digital systems that support coherent behavior by making participants and their activities visible to one another". Designing social infrastructure aims at supporting *mutual awareness and accountability* and thus facilitating participation and collaboration. Interpretation and understanding of those actions can influence the direction of public decision making [15], influence reputations [16], notions of expertise [17], as well as other aspects of collaboration.

12.4 The Tension Between Privacy and Social Translucence

In the previous section we saw that visibility is a crucial requirement to sustain participation and accountability. However, we speak of socially translucent systems rather than socially transparent systems, because there is a vital tension between privacy and visibility [9]. Indeed, sensing from a cellphone for collecting information from the environment and tagging them with time and GPS data, could be used to infer a lot of personal information, including the user's identity. This problem is often termed *location privacy*. Knowing when a particular person was at a particular point in time can be used to infer the personal activities, political views, health status, and launch unsolicited advertising, physical attacks or harassment. Location information is therefore in many cases a particularly sensitive piece of personal data and people have now started to realize more and more the need for location privacy [10].

In our pessimistic scenario users have strong concerns about privacy, while at the same time we would like to preserve social features as much as possible. What kind of social translucence design can we offer in this context? Is it possible to offer anonymity to the user, who submits sensing data from the physical environment, while at the same time we maintain properties connected to the translucence of his online identity, like reputation and accountability? So, in a way, while at the previous section we were looking to bring the users from the physical

environment closer to the online communities, now we seek ways to maintain these benefits, but also separate their physical identity from their online identity.

In the following sections, we try to answer these questions by showing that indeed, (i) it is possible to revoke access credentials of anonymous misbehaving users and also (ii) award reputation points to anonymous users submitting data. In the latter case reputation values would be public, appearing on the user public profile in the community to express the degree of his/her contribution and receive attention.

12.4.1 System Architecture

As it turns out, it is not a trivial task to provide anonymity for pervasive contributions of sensor data, as many actors are involved in the process, who could potential harm the privacy of the users. Let us assume that any identifying information has been removed from the data, so it includes only the sensing information, the GPS value and the time of measurement. This is not enough to provide anonymity to the user, if we do not first of all protect identifying information at the network layer. Network identifiers can be used, either to reveal the identity of the user directly or link several reports back to the same user and therefore build a location profile of that user.

Figure 12.2 depicts the communication paths between the two communication ends in a generic participatory sensing architecture: the mobile users and the application provider. There are (at least) two network access possibilities for the user: through a data telecommunications service, like GSM or UMTS and through a (possibly open) WLAN access point. Providing anonymity at the first hop of communication, i.e. between the user and the mobile operator or the Wi-Fi hotspot, is a problem that falls outside the scope of this chapter. Here we consider attackers, who are able to observe the traffic over the Internet between the access point and the service provider. At this level the goal is to provide communication anonymity, which means hiding the network identifiers in the network layer (i.e., IP addresses).

Since mixes were proposed in 1981 [18] as a solution for achieving anonymous communication, multiple other protocols appeared in the literature in order to provide anonymity over the Internet. In particular, low-latency anonymous overlay networks seek to provide, from the user's point of view, a reasonable trade-off between anonymity and performance. Some of the most prominent low-latency approaches include Crowds, Tor, Jap, and Onion Routing. Still, only a few of these anonymizing networks have been tested for the mobile Internet scenario and it is an area that only lately attracted research interest [19]. Even though it is not hard to adapt protocols like Tor to conform to the mobile internet constraints, other more lightweight solutions remain to be investigated [20]. Nevertheless, in the rest of this book chapter, we will assume that a suitable anonymous overlay network is applied to offer the desirable protection at the communication level.

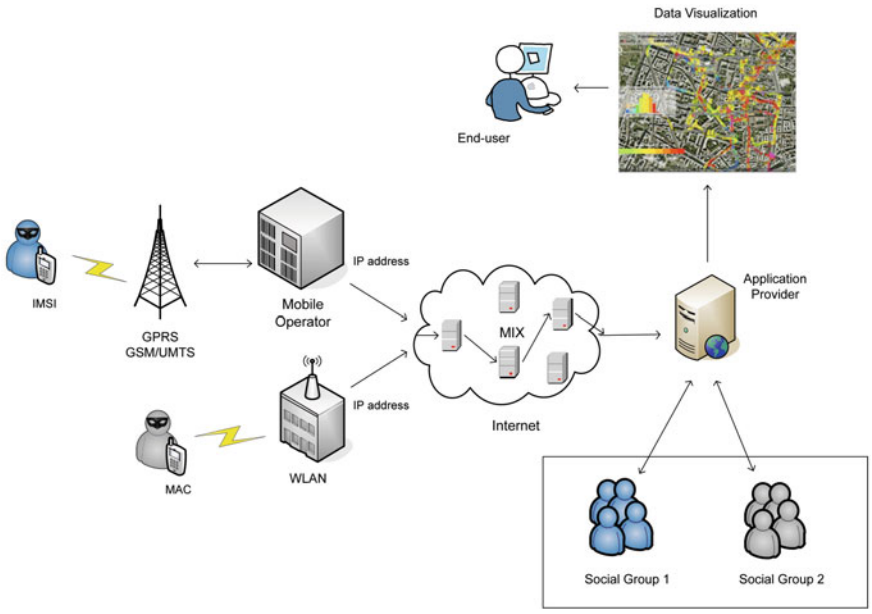


Fig. 12.2 A generic system model for a participatory sensing system, where users preserve their location privacy by submitting de-identified data. However, the network layer contains many identifiers that can be used to identify the users. Also the knowledge of the social group that an anonymous user belongs to, could reduce the offered anonymity

Let us note however that the interconnection of users through online communities creates a different setting for the evaluation of the performance by anonymous communication networks in our scenario. Here, an attacker, besides her observations at the network layer, has also knowledge from the application layer, i.e., the identities of the users that participate in the system and how they are related, through their profiles in the social group. Users organize themselves into a community with a common goal, and these users are expected to send measurements for the corresponding campaign. There is an a priori knowledge of user profiles and associations that can be combined with data gathered by traffic analysis of the mix-based network.

Diaz et al. studied the problem of measuring anonymity based on profile information [21] and social networks [22] and showed that user profile information does not *necessarily* lead to a reduction of the attacker's uncertainty. The assumptions in this work include a 1-to-1 communication paradigm, where individuals communicate with each other directly, as well as a global passive adversary model, where the attacker can observe all the inputs and outputs of the anonymous communication network. Generalizing the first and relaxing the second assumption certainly creates an interesting but also challenging problem.

12.4.2 Revocation of Misbehaving Users

For most participatory sensing applications it is essential to enforce access control in order to prevent service abuse and to protect against malicious attacks. Access to services for users offering the data should be granted only based on pre-established trust between users and the service provider. Given that we also want to preserve anonymity, this leads to a chicken-and-egg conundrum. On one hand, a user has to be authenticated before accessing a service; on the other hand the users ID can serve as a unique identifier that can be used to track the users whereabouts and actions.

In response to this problem, a lot of research work has focused on anonymous user authentication that targets user privacy while maintaining access security. The basic idea has been to verify the users right to access a service, while at the same time the users identifying information remains secured. This immediately creates an important requirement: the support of *user revocation*. The anonymous access to a service offers users a high degree of privacy and along with it the license to misbehave without the fear of punishment. Therefore we want to be able to de-anonymize misbehaving users and limit their access to the system.

An approach for enhancing anonymous authentication is to use *group signatures* [23], where a vast amount of research is being carried out worldwide. These technologies can be used to verify whether or not a user is allowed access, without actually identifying the user. This is achieved by allowing a member of a group to sign a message on behalf of the group, without revealing which member produced the signature. Group signature systems can support revocation, where group membership can be selectively disabled without affecting unrevoked members.

In order to apply group signatures for mobile phones and users belonging to highly dynamic communities, we need to address a number of problems that come with this solution. For example, in online communities members continuously come and go and a solution to change and re-distribute fresh certificates to all members each time is not a viable solution. This problem has been addressed by anonymous credential systems that support dynamic membership revocation [24, 25].

Existing group signature solutions are based on a trusted third party (TTP), which has the ability to revoke a user's privacy at any time. This becomes problematic, since users can never be assured that their privacy will be maintained by that TTP. To eliminate the reliance on TTPs, certain "threshold-based" approaches such as e-cash [26, 27] and k-Times Anonymous Authentication (k-TAA) [28] have been proposed. In these schemes, no one, not even an authority, can identify a user who has not exceeded the allowed number of k authentications or spent an e-coin twice.

However, misbehavior in participatory sensing applications is not defined as overusing a service. In our case, we are interested in revoking users who upload data, which after a specific process are judged as "inappropriate". When they have been judged to have repeatedly misbehaved at least d times, they should be revoked by the system. This problem has been addressed recently by Tsang et al.

[29], who proposed a *d*-strikes-out revocation scheme for blacklisting misbehaving users, without relying on a TTP. Unfortunately the computational and communication overhead of the protocol is not attractive for power-limited devices such as mobile phones, especially as the size of the blacklist grows.

12.4.3 Anonymous Reputation

As we discussed above, offering reputation points to people submitting data can form a sort of recognition to their efforts. These reputation points are collected when submitting data to the service provider and then they are publicly displayed in the profile that the user maintains in the community. The challenge to comply with the privacy properties that we also described above should now be obvious. A direct process of acquiring reputation points for a given report and display them on a public profile would clearly compromise the anonymity of the submitter. So, we need to provide a protocol that satisfies the following two properties:

- The process of acquiring reputation points is independent from the process of updating the reputation value on someone's public profile.
- The process of acquiring reputation points for two successive reports should be unlinkable with each other, in order to maintain the unlinkability of reports.

One way is to base the solution on Chaum's eCash [18]. An electronic cash system aims at emulating regular cash and offers anonymity properties: an adversary cannot link a spending to a withdrawal. In our system, the whole process takes place in two independent phases: First a user U communicates with the service provider under a randomly chosen one-time pseudonym P_U to submit the data. The user obtains an e-coin from the bank for each report submission, each one corresponding to a reputation point. In the second phase, the user logs-in using his regular public profile and redeems the e-coin to get a reputation point and increase his total reputation. E-coins can be spend only once, and cannot be transferred to other users.

However, we cannot solve the problem of a secure reputation system just by using an eCash scheme. E-coins are anonymous, but linkable, which in turn leads to the linkability of the reports submitted in order to acquire these e-coins. The use of other cryptographic tools are required as well, such as blind signatures [30]. For an example of how these cryptographic tools can be combined to build a reputation system for anonymous networks, we refer the reader to the recent work of Androulaki et al. [31]. One of the drawbacks of this protocol is that negative reputation is not supported. That is, users can only increase their reputation and eventually the system will reach a final state, where all users have the maximum reputation. After this point, no user has the incentive to collect new reputation points.

To address this problem, Schiffner et al. [32] proposed a solution that supports non-monotonic reputation. By allowing negative ratings, the problem that emerges is that ratees cannot be forced to deposit received reputation coins, i.e., the ratee

can decide on his own whether he wants to deposit the received rating and of course he would not deposit a negative coin. To overcome this, the authors force the rating of every interaction. That is, the reputation provider keeps account not only of the reputation, but also of the interactions, guaranteeing that each interaction is actual rated, possibly also in a negative way.

12.5 Conclusions

In this book chapter we introduced the emerging research area of participatory sensing and concentrated on the location privacy challenges. We took the approach of protecting user's privacy by anonymizing their data before submission to the service provider. On the other hand, we argued that offering social translucence to the users is important for the success of future participatory sensing campaigns. As these two goals are contradictory with each other, we looked for social translucence properties that are still possible, even under user anonymity like accountability and reputation.

For the first, we saw that indeed revoking anonymous misbehaving users is possible, even without relying on a trusted third party (TTP), but more work would be needed to improve the performance of such protocols in the pervasive scenario. For the latter, some protocols exist that allow anonymous users to collect reputation points using a combination of e-cash systems and blind signatures. However, they currently do not support more complicated reputation systems that will be needed to support incentive and community building mechanisms of participatory sensing. Finally, it remains an interesting problem to see what other tools we can develop in the future to provide even more social translucence for anonymous users in participatory sensing systems.

Acknowledegments We are grateful to Felix Freiling for the fruitful discussions regarding this work.

References

1. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., Srivastava, M.B.: Participatory sensing. In: Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications, pp. 117–134, October (2006)
2. Irwin, A.: Citizen science: a study of people, expertise and sustainable development. Routledge, London (1995)
3. Paulos, E., Honicky, R., Hooker, B.: Citizen science: enabling participatory urbanism. In: Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City. IGI Global, Hershey (2008)
4. Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., Peterson, R., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G.-S.: The rise of people-centric sensing. IEEE Internet Comput. **12**(4), 12–21 (2008)

5. Cuff, D., Hansen, M., Kang, J.: Urban sensing: out of the woods. Commun. ACM **51**(3), 24–33 (2008)
6. Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.: A survey of mobile phone sensing. IEEE Commun. Mag. **48**, 140–150 (2010)
7. Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T.: Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08), pp. 337–350 (2008)
8. Nicolas Maisonneuve, M.S., Ochab, B.: Participatory noise pollution monitoring using mobile phones. Inf. Policy **15**, 51–71 (2010)
9. Erickson, T., Kellogg, W.A.: Social translucence: an approach to designing systems that support social processes. ACM Trans. Comput.–Hum. Interact. **7**(1), 59–83 (2000)
10. Krontiris, I., Freiling, F.C., Dimitriou, T.: Location privacy in urban sensing networks: research challenges and directions. IEEE Wireless Commun. Mag. **17**, 30–35 (2010)
11. Paxton, M.: Participate: producing a mass scale environmental campaign for pervasive technology. In: 6th International Conference on Pervasive Computing (2008)
12. Kollock, P.: The economies of online cooperation: gifts and public goods in cyberspace. In: Communities in Cyberspace, pp. 220–239 (1999, Chap. 9)
13. Aguiton, C., Cardon, D.: The strength of weak cooperation: an attempt to understand the meaning of web 2.0. Commun. Strategies **65**, 51–65 (2007)
14. Bryant, S.L., Forte, A., Bruckman, A.: Becoming wikipedian: transformation of participation in a collaborative online encyclopedia. In: GROUP '05: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, pp. 1–10. ACM, New York (2005)
15. Borning, A., Friedman, B., Davis, J., Lin, P.: Informing public deliberation: value sensitive design of indicators for a large-scale urban simulation. In: Proceedings of the 9th European Conference on Computer-Supported Cooperative Work (2005)
16. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Commun. ACM **43**(12), 45–48 (2000)
17. Mcdonald, D., Ackermann, M.: Just talk to me: a field study of expertise location. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (1998)
18. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)
19. Lenhard, J., Loesing, K., Wirtz, G.: Performance measurements of Tor hidden services in low-bandwidth access networks. In: Proceedings of the International Conference of Applied Cryptography and Network Security (ACNS '09), pp. 324–341, June (2009)
20. Krontiris, I., Freiling, F.C.: Integrating people-centric sensing with social networks: a privacy research agenda. In: Proceedings of the IEEE International Workshop on Security and Social Networking (Sesoc) (2010)
21. Diaz, C., Troncoso, C., Danezis, G.: Does additional information always reduce anonymity? In: Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society (WPES '07), pp. 72–75 (2007)
22. Diaz, C., Troncoso, C., Serjantov, A.: On the impact of social network profiling on anonymity. In: Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (2008)
23. Chaum, D., van Heyst, E.: Group signatures. In: Advances in Cryptology—EUROCRYPT '91, pp. 257–265 (1991)
24. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Advances in Cryptology—CRYPTO 2002, pp. 61–76. Springer, London (2002)
25. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04), pp. 168–177 (2004)

26. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Advances in Cryptology—EUROCRYPT 2005, pp. 302–321 (2005)
27. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Balancing accountability and privacy using e-cash (extended abstract). In: Proceedings of the 5th Conference of Security and Cryptography for Networks (SCN '06), pp. 141–155 (2006)
28. Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication. In: Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '04), pp. 308–322, December (2004)
29. Tsang, P.P., Au, M.H., Kapadia, A., Smith, S.W.: BLAC: revoking repeatedly misbehaving anonymous users without relying on TTPs (2010)
30. Chaum, D.: Blind signature system. In: CRYPTO, p. 153 (1983)
31. Androulaki, E., Choi, S.G., Bellovin, S.M., Malkin, T.: Reputation systems for anonymous networks. In: Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (PETS '08), Leuven, Belgium, pp. 202–218 (2008)
32. Schiffner, S., Clauß, S., Steinbrecher, S.: Privacy and liveliness for reputation systems. In: Proceedings of the 6th European PKI Workshop: Research and Applications (EuroPKI '09), Pisa, Italy (2009)

Chapter 13
A Summary of Two Practical Attacks Against Social Networks

Leyla Bilge, Marco Balduzzi, Davide Balzarotti and Engin Kirda

Abstract Social networking sites have been increasingly gaining popularity, and they have already changed the communication habits of hundred of millions of users. Unfortunately, this new technology can easily be misused to collect private information and violate the users' privacy. In this chapter, we summarize two practical attacks we have presented in the past: an impersonation attack in which we automatically clone a user profile, and an attack that abuses the information provided by social networks to automatically correlate information extracted from different social networks. Our results show that these attacks are very successful in practice and that they can significantly impact the users' privacy. Therefore, these attacks represent a first important step to raise awareness among users about the privacy and security risks involved in sharing information in one or more social networks.

Keywords Social networks · Security attack · Impersonation · Correlation

L. Bilge (✉) · M. Balduzzi · D. Balzarotti · E. Kirda
Institute Eurecom, Valbonne, Sophia Antipolis, France
e-mail: bilge@eurecom.fr

M. Balduzzi
e-mail: balduzzi@eurecom.fr

D. Balzarotti
e-mail: balzarotti@eurecom.fr

E. Kirda
e-mail: kirda@eurecom.fr

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_13,

13.1 Introduction

A social network is a social structure that is made up of nodes representing individuals or organizations. These nodes may be tied to each other by properties such as friendship, common values, visions, ideas, business relationships and general interests. Although the idea of social networks has been around for a long time (e.g., see [1]), social networking web sites and services are a relatively new phenomenon on the Internet. Business relationship-focused social networking sites such as XING [2] (previously known as OpenBC) and LinkedIn [3], as well as friendship-focused social networking sites such as Facebook [4], MySpace [5], StudiVZ [6] and MeinVZ [7] have been gaining popularity among Internet users. In fact, LinkedIn boasts on its web site that it has 30 million registered users. XING, a business networking site that is very popular in Switzerland, Germany and Austria, claims to have 6 million registered users. Although it has only been created four years ago, Facebook now has more than 400 million active users and is reporting growth rates of 3% per week. According to Facebook, it registers 30 billion page views per month and is the largest photo storage site on the web with over 1 billion uploaded photos [8].

With the introduction of social networks, the Internet community experienced a revolution in its communication habits. What initially began as a simple frame for social contacts quickly evolved into massively-used platforms where networking and messaging is only one of the multiple possibilities the users can call upon. While basic messaging is still one of the key features, it is clear that the participants see the main advantage in the well-organized representation of friends and acquaintances.

Unfortunately, as the interest for a new technology grows on the Internet, miscreants are attracted as well. For example, spam was not a major problem until the end of the 1990s. However, as more and more people started using e-mail, unsolicited (i.e, spam) e-mails started increasing in numbers. In fact, spam has reached such high proportions that the Spamhouse Project [9] now estimates that about 90% of the incoming e-mail traffic in North America, Europa and Australasia is spam. Also, the increase in the popularity of e-mail also resulted in an increase in the number of malicious e-mails (e.g., e-mails with worm attachments, phishing e-mails, scam e-mails, etc.). Today, e-mail is a popular way of spreading infections.

As the popularity of social networking sites increase, so does their attractiveness for criminals. For example, worms have recently emerged that specifically target MySpace and Facebook users [10]. These worms make use of old ideas that are applied to a new technology. Analogous to classic worms such as LoveLetter [11] that used the contacts in a victim's Outlook address book to spread, these new social networking worms use the friend lists of a victim to send a copy of themselves to other social networking users. Although such e-mail attachments may raise more suspicion now as such tricks have already been seen by many e-mail users, they are not as well-known on social networking sites. Furthermore, note that incoming e-mails with attachments are often scanned for malicious content and Bayesian filters are applied to sort out unsolicited mails. In

comparison, social networking sites do not usually provide filtering mechanisms or warnings for dangerous content, hence, making it easier, in principle, for a potential attacker to send malicious applications and URLs to victims.

Social networking sites are an attractive target for attackers because of the nature of the sensitive information that they contain on registered users. Typically, users enter their real e-mail addresses and provide information on their education, friends, professional background, activities they are involved in, their current relationship status and sometimes even list their previous relationships (e.g., on Facebook, one may read that Mr X. was together with Ms Y until they broke up in 2006). Hence, from the attacker's point of view, access to this type of detailed, personal information would be ideal for launching targeted, social engineering attacks, now often referred to as spear phishing [12, 13]. Furthermore, the collected e-mail addresses and personal information would be invaluable for spammers as they would (1) have access to e-mail addresses that belong to real people (i.e., one problem spammers face is that they often do not know if the e-mail addresses that they collect are indeed being used by real people or they are just secondary addresses that are not regularly read) and (2) have information about the people using these e-mail addresses allowing them to efficiently personalize their marketing activities, tailored according to the knowledge from the target's profile. Also, note that the ability to associate personal information with an e-mail address is important to be able to successfully by-pass spam filters [14]. Such filters usually generate a list of "spammy" tokens versus "good" tokens after training with a large set of previously received e-mails. As a result, e-mails that contain the name of the user receiving the e-mail, or names of people that he is acquainted with tend to receive lower spam ratings than e-mails that are less personal. As a result, if the spammer is able to include some personal information in the spam that he is sending, he would be able to improve his chances of reaching the targeted user.

For a social networking site to work properly, it is imperative to have certain knowledge about the participants. Suggesting users from the same area with the same age, for instance, can lead to a renewed childhood friendship, while a detailed work history might open unexpected business opportunities. On the other hand, this kind of information is also of great value to entities with potentially malicious intentions. Hence, it is the responsibility of the service provider to ensure that unauthorized access to sensitive profile information is properly restricted. In fact, various researchers (e.g., [15–17]) have shown that social networks can pose a significant threat to users' privacy as well. The main problem is twofold:

- Many users tend to be overly revealing when publishing personal information. Although it lies in the responsibility of each individual to assess the risk of publishing sensitive information, the provider can help by setting defaults that restrict the access to this information to a limited number of individuals. A good example is Facebook, where detailed information is only exchanged between already connected users.
- Information exists in social networks that a user cannot directly control, and may not even be aware of. The best example is the use of the information

provided during the registration phase (e.g., name, contact e-mail address, and birthday). Even though this data may never be shown in the public user profile, what most users do not realize is the fact that this information is still often used to provide other functionality within the social network (e.g., such as determining which users might know each other).

In this chapter, we give a summary of a number of practical attacks that were studied in the past [18, 19]. First, two impersonation attacks that consist of the automated identity theft of real user profiles are explained. Then, a type of attack that abuses the information provided by social networks for automated user profiling is described.

In the first impersonation attack, an already existing profile in a social network is cloned and friend requests are sent to the contacts of the victim. Hence, the contacts of a user can be "stolen" by forging his identity and creating a second, identical profile in the same social network. Having access to the contacts of a victim, therefore, means that the sensitive personal information provided by these contacts can be accessed. The experimental results show that a typical user tends to accept a friend request from a forged identity who is actually already a confirmed contact in their friend list.

In the other one, we perform a *cross-site* profile cloning attack where we clone a profile that exist in one social network, but not in the other. In this attack, users who are registered in one social network, but who are not registered in another are automatically identified. This makes it possible to clone the identity of a victim in the site where he is registered, and forge it in a social networking site where he is not registered yet. After the forged identity is successfully created, then the social network of the victim is rebuilt by contacting his friends that were identified to be registered on both social networking sites. The experimental results suggest that this attack is especially effective because profiles in this case only exist once on the social networking site that is being targeted. As a result, the friend requests that are sent look perfectly legitimate and do not raise suspicion with the users who have been contacted.

The attack that performs automated user profiling exploits a common weakness shared by eight most popular social networks: Facebook, MySpace, Twitter, LinkedIn, Friendster, Badoo, Netlog, and XING. The weakness is inherent in a feature that is particularly useful for newly-registered users: *Finding friends*. With the functionality to search for friends, social networks need to walk the thin line between revealing only limited information about their users, and simplifying the process of finding existing friends by disclosing the personal details of registered users. A common functionality among these popular social networks is to let users search for friends by providing their e-mail addresses. For example, by entering "gerhard@gmail.com", a user can check if her friend Gerhard has an account on the social network so that she can contact and add him to her friend list. Note that an e-mail address, by default, is *considered to be private information*, and social networks take measures *not to reveal this information*. That is, one cannot typically access a user's profile and simply gain access to his personal e-mail address.

One of the main purposes of protecting e-mail addresses is to prevent spammers from crawling the network and collecting e-mail to user mappings. With these mappings at hand, the attacker could easily construct targeted spam and phishing e-mails (e.g., using real names, names of friends, and other personal information [20]). This kind of profiling is also interesting for an attacker to perform a reconnaissance prior to attacking a company. By correlating mappings from different social networks, it is even possible to identify contradictions and untruthfully entered information among profiles.

13.2 Cloning Attacks Against Social Networking Sites

13.2.1 Profile Cloning

Our premise for the profile cloning attack is that social networking users are generally not cautious when accepting friend requests. Our assumption, as an attacker, is that many users will not get suspicious if a friend request comes from someone they know, even if this person is already on their contact list. In fact, some users may have hundreds of confirmed contacts in their friend lists and they may have varying levels of communication with these people. For example, one might exchange messages with a primary school friend once a year, but have daily contact with a friend who is in the same city. Because of the lower degree of contact, the chance that the primary school friend will get suspicious for the new duplicate contact request is less than someone the victim is in regular contact with.

Typically, whenever a user receives a friend request, she needs to confirm the relationship and accept the new connection. Either a standard friendship message can be sent, or a personal message can be added to it. For example, to make the new friend request more convincing, the attacker may add a social engineering message such as "Dear friends, my computer broke down, I am reconstructing my friend list. Please add me again!". A real attacker likely prefers to use a personal message to increase her success rate.

Of course, it is likely that after a while the victims will notice the abnormality in their friend list and will remove the fake friend. Even though this seems to be undesirable, from the attacker's point of view it is enough for a contact to accept a friend request. Even if the contact decides to remove the friend connection later on, the attacker already had a chance to access and copy the victim's personal information.

The profile cloning attack consists of identifying a victim and creating a new account with his real name and photograph inside the same social network. The profile photographs can be simply copied and used when registering a new, cloned account. Furthermore, note that names are not unique on social networks, and people may exist who have identical names.

Once the cloned account has been created, our system can automatically contact the friends of the victim and send friend requests. Whenever a user receives a friend request, she typically sees the photograph and the name of the person who

has sent the request. Our expectation, as the attacker, is that the user will accept this request as it is from someone they recognize and know.

In our experiments, we measured the feasibility of the cloning attack on Facebook [18]. The results show that average friendship acceptance rate for forged profiles is over 75%. We believe that an attacker that performs such an attack is able to gain the trust of the friends of the forged profile.

13.2.2 Cross-site Profile Cloning

In the cross-site profile cloning attack, our aim is to identify victims who are registered in one social network, but not in another. Our first aim, from the attacker's point of view, is to steal their identities and create accounts for them in the network where they are not registered. Note that this attack is more difficult to detect by the social network service provider or the legitimate owner of the copied profile. As far as the service provider is concerned, a new user is registering to the network.

When creating an identical account in another social network, we attempt to retrieve as much information as possible from the victim's original account in the other network. Clearly, the type of the social network is relevant when forging accounts. That is, it is much easier for an attacker to create forged accounts in social networks of the same nature.

Our second aim, after the stolen identity has been created, is to identify the friends of the victim in the original network and check which of them are registered in the target network. To determine with certainty if a friend of the cloned contact is already registered on a different social network is not as straight-forward as it may seem. In order to determine with a high probability if a certain user already exists on a social network, we need to look at more information associated with that specific user. In our comparison, the information that we take into account consists of the name, the educational background, the professional background of the profile, and finally, the location where the user lives.

Once the contacts of a victim have been identified, our system can then start sending automated friend requests to these identified users. As far as the contacted users are concerned, a friend request is coming from someone they know and who is not on their friend list yet. As a result, our expectation is that most users will accept this request without becoming suspicious. After all, it is the nature of social networks that people get connected by receiving friend requests from time to time from people that they know.

We applied cross-site profile cloning attacks to XING and LinkedIn since they both focus on business connections. Our experiments suggest that the cross-site profile cloning attack is more effective in practice than the profile cloning attack. This is because profiles exist only once on the social networking site that is being targeted. As a result, the friend requests that we send look perfectly legitimate and do not raise suspicion with the users who have been contacted.

13.3 Abusing Social Networks for Automated User Profiling

Many social network providers such as Facebook, MySpace, XING, or LinkedIn offer a feature that allows a user to search for her friends by providing a list of e-mail addresses. In return, the user receives a list of accounts that are registered with these e-mail addresses. From a user's point of view, this feature is valuable: A user can simply upload her address book, and the social network tells her which of her friends are already registered on the site. The feature enables a user to quickly identify other users she knows, and with which she might be interested in establishing a connection.

While the e-mail search functionality commonly available in social networks is convenient, a closer examination reveals that it also has some security-relevant drawbacks. An attacker can misuse this feature by repeatedly querying a large number of e-mail addresses using the search interface as an oracle to validate users on the social network. This information can then be abused in many ways, for example:

- A spammer can automatically validate his list of e-mail addresses (e.g., find out which addresses are most probably real and active) by querying a social network, and only send spam e-mails to those users [21].
- The previous attack can be combined with *social phishing*, i.e., the spammer crawls the profile of a user and uses this information to send targeted phishing e-mails (if the user has a public profile and a public friend list) [20].
- An attacker can generate detailed profiles of the employees of a company and use this information during the reconnaissance phase prior to the actual attack.

Note that it has been recently reported that spammers have started to shift their attention to social networking sites to collect information about users that they can then use for targeted e-mails [22]. The report states that spammers have been using bots to spy information from social networks that they can then use for launching attacks such as guessing passwords (i.e., using reminder hints such as "What is my favorite pet?"). The prerequisite for these current attacks, however, is that a bot is installed on the victim's machine. In comparison, we describe the exploitation of a common weakness in a social network functionality that allows us to retrieve information about users even if they are not infected by a bot.

In each of these cases, the attack is only feasible since the social network provider enables a large-scale query of e-mail addresses. Before going into details on how this feature can be abused in practice, we provide an overview of the context of this type of attacks and previous instances of similar problems.

Once the attacker is able to abuse the e-mail search functionality to validate users on the social network (see Fig. 13.1a), in the second step, the attacker retrieves the user's profile from the different networks in an automated way (see Fig. 13.1b). From each profile, she extracts the information she is interested in, for example, age, location, job/company, list of friends, education, or any other

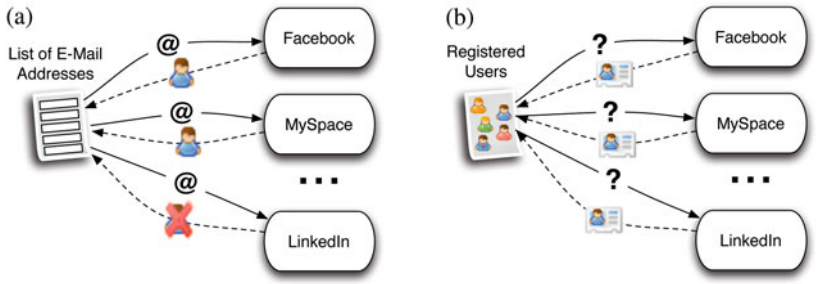


Fig. 13.1 Automated user profiling based on information collected on social networks. **a** Querying social networks for registered e-mail addresses on a scale. **b** Crawling every profile found in the first step to collect personal information

information that is publicly available. This information can then be aggregated and correlated to build a rich user profile.

This attack can be realized even with very limited resources. In fact, by using a single machine over a few weeks only, it is possible to collect hundreds of thousands of user profiles, and queries for millions of e-mail addresses (i.e., each social network was successfully queried for 10.4 million addresses, adding up to a total of about 82.3 million queries). This emphasizes the magnitude and the significance of the attack since a more powerful, sophisticated, and determined attacker could potentially extract even more information (e.g., by using a large botnet).

An attacker can also abuse the search feature in a completely different way, extending the attack. During the profiling step, an attacker can learn the names of a user's friends. This information is often available publicly, including social networking sites such as Facebook and Twitter. An attacker can thus obtain the tuple (first name, last name) for each friend of a given user, but not the e-mail addresses for these friends: The e-mail address itself is considered private information and not directly revealed by the social networking sites. However, an attacker can automatically try to guess the e-mail addresses of the friends of a user by abusing the search feature. We implemented two different, straight-forward techniques for generating new e-mail addresses, based on user names.

For the first technique, for each friend, we build 24 addresses. Given a name in the form "*claudio bianchi*", we generate six prefixes as "*claudio.bianchi*", "*claudiobianchi*", "*claudio_bianchi*", "*c.bianchi c_bianchi*" and "*cbianchi*". Then, we append the four most popular free e-mail domains "*gmail.com*", "*yahoo.com*", "*aol.com*", and "*hotmail.com*".

For the second technique, we use context information for generating e-mail addresses: If a user has an e-mail address with a certain structure (e.g., automatically generated e-mail accounts often include the last name of the user and a static prefix), we try to detect this structure by searching the user's first and last name within the e-mail address. If we identify a pattern in the address, we use this match

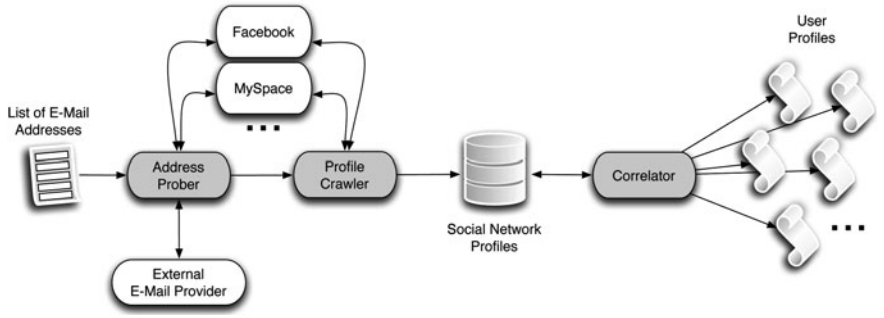


Fig. 13.2 Overview of system architecture

and generate two additional e-mail addresses that follow the same pattern (including both the first and last name) for each friend. If we do not detect a pattern, we generate e-mail addresses similar to the first algorithm. However, instead of appending common prefixes, we use the prefix of the user on the assumption that the friends of a user might be a member of the same e-mail provider.

Finally with this approach, an attacker can generate new e-mail addresses based on profile information, and brute-force the social networking sites to validate them.

13.3.1 Implementation of the Attack

Our prototype system [19] has been implemented as a collection of several components. One component queries the social networks, one extracts and stores the identified information from user profiles, and one automatically correlates the information to discover as much information as possible about a user. An overview of the system and the relationship of the components is shown in Fig. 13.2.

We designed our system to be efficient and stealthy at the same time. Therefore, we had to find a compromise between normal user behavior, which is stealthy, and brute-force crawling, which is efficient but bears the danger of frequently-suspended accounts. Our solution was tweaked for each social network, to find the right combination of timeouts and number of requests. Furthermore, our solutions was carefully designed not to overwhelm the tested networks.

Our architecture contains three main components: the *Address Prober*, the *Profile Crawler*, and the *Correlator*.

The *Address Prober* is an HTTP client that is responsible for uploading the list of e-mail addresses to be queried to the social network. The social network, in return, sends back the list of accounts that are registered with those addresses. The Address Prober also supports external e-mail providers such as, for example, Google's webmail service Gmail, and permits to upload lists of e-mail addresses to such accounts. The motivation behind this feature is that some social networks

only support e-mail queries if the source is an external e-mail account with an attached address book.

The *Profile Crawler* is responsible for a deeper investigation of the user profiles discovered in the previous step. The goal is to gather as much information about a single user as possible.

After the crawling phase, the *Correlator* component combines and correlates the profiles that have been collected from the different social networks. The email address is used as a unique identifier to combine together different profiles and identify the ones that belong to the same person. When it finds two profiles associated with the same e-mail address, the Correlator compares all the information in the two profiles to identify possible inconsistencies.

In particular, it compares all the fields that can assume a small set of values, e.g., sex (either male or female), age (a positive integer number), and current relationship (married, single, or in a relationship). The correlation phase has two main goals: first, if a person provides his full name in social network *A*, but registers a profile in the network *B* using a pseudonym, by cross-correlating the two profiles, we can automatically associate the real user's name also to the account *B*. Second, we can detect inconsistent values across multiple networks. For example, Bob can provide his real age on his profile on social network *A*, while pretending to be 10 years younger on social network *B*.

In our experiments, we used about 10.4 million real-world e-mail addresses that were left by attackers on a dropzone on a compromised machine. We built a system to automatically query eight social networking sites with these addresses, just as an adversary would, and we were able to identify about 1.2 million profiles that were linked to one of the e-mail addresses we probe. Most profiles were found on Facebook (4.96%), LinkedIn (2.36%), and MySpace (2.01%). Our experiments demonstrated that we were able to automatically extract information about users that they might actually wish to hide certain online behavior. For example, we could identify users who were potentially using a different name on a dating web site, and were pretending to be younger than they really were. The correlation that we were able to do automatically has a significant privacy impact.

13.4 Related Work

Social networks comprise of nodes that are connected to each other via strong trusted links. That is, they rely on the assumption that a significant fraction of the users in the system are honest. The most well-known attack to compromise the trust relationship in a social network that employs a reputation system is the *sybil attack* [23]. In this attack, the attacker creates multiple fake identities and pretends to be distinct users in the network, using them to gain a disproportionately large influence on the reputation system.

To date, in order to defend social networks against sybil attacks, two systems were proposed: SybilGuard [24] and SybilLimit [25]. The key insight used in both approaches is that real-world social networks are fast mixing [26, 27] that aids to distinguish the sybil nodes from normal nodes. Fast mixing means that subsets of honest nodes have good connectivity to the rest of the social network.

Both SybilGuard and SybilLimit are good solutions for detecting Sybil nodes. However, in our profile cloning attacks the established friendship connections are legitimate and the system is establishing contact to a high number of existing "honest" nodes. Therefore, our fake accounts would not be detected by the previous approaches.

A study that is very related to the experiments we presented for profile cloning attacks was conducted by Sophos [28]. The authors created a profile on Facebook [4] and manually sent friend requests to 200 random users. The study reports that 41% of the users accepted the request. Furthermore, most of the users did not restrict the access to the personal information in their profile. Note that the results of our experiments are consistent with the study conducted by Sophos and demonstrate that many users are not cautious in social networks. However, one of the main differences between our work and the experiment performed by Sophos is that we are able to automatically identify target users and send friend requests and we show how the attack success rate can be greatly improved by cloning real user accounts.

In [20], the authors present experiments that they have performed on "social phishing". They have crawled a number of social networking sites and have downloaded publicly available information on users. Then, they manually constructed phishing e-mails that contained some personal information on the victims that they were able to retrieve from the social networking sites. The results of the study show that victims are more likely to fall for phishing attempts if some information about their friends or about themselves is included in the phishing mail. Our results, without relying on email messages, confirm that there is a high degree of trust in social networks. However, our focus is different as we aim at accessing the personal information of users that have not necessarily made their profile public.

The large popularity of social networks and the availability of large amounts of personal information has been unprecedented on the Internet. As a result, this increasing popularity has lead to many recent studies that examine the security and privacy aspects of these networks (e.g., [17, 18, 20, 21, 29–32]). As more and more Internet users are registering on social networking sites and are sharing private information, it is important to understand the significance of the risks that are involved.

The structure and topology of different social networks was examined by different research groups (e.g., [33–36]). The main focus of previous work was either on efficient crawling or on understanding the different aspects of the graph structure of social networks. We extend previous work by contributing a novel way to enumerate users on social networks with the help of e-mail lookups.

The automated user profile is facilitated by the fact that an attacker can use an e-mail address to link profiles on different social networks to a single user.

The idea of correlating data from different sources to build a user profile has been studied in different contexts before. For example, Griffith and Jakobsson showed that it is possible to correlate information from public records to better guess the mother's maiden name for a person [37]. Heatherly et al. [38], and Zheleva and Getoor [36] recently showed that hidden information on a user's profile can also be inferred with the help of contextual information (e.g., the political affiliation of a user can be predicted by examining political affiliation of friends).

Concurrently and independently of the work on automated user profiling, Irani et al. [39] performed a similar study of social networks. They showed that it is straightforward to reconstruct the identify (what they call the *social footprint*) of a person by correlating social network profiles of different networks. The correlation is done either by using the user's pseudonym or by inferring it from the user's real name. In contrast, our work focuses on automated techniques to find profiles of the same person on different networks. In fact, due to the friend-finder weakness that we discovered on all tested networks, we are able to associate profiles by e-mail addresses. As a result, we produce a more precise correlation: On one hand, we can make sure that different profiles belong to the same individual (Irani et al. have a positive score of only 40% for the pseudonym match and 10–30% for the real name match). On the other hand, we can reveal the "'hidden profiles" of users that they may actually wish to hide. Indeed, this is a major advantage of our approach; we can link profiles that are registered using different pseudonyms or information, but based on the same e-mail address.

Also, note that our work is also related to the area of *de-anonymization*, where an attacker tries to correlate information obtained in different contexts to learn more about the identity of a victim. Narayanan and Shmatikov showed that by combining data with background knowledge, an attacker is capable of identifying a user [40]. They applied their technique to the Internet movie database (IMDb) as background knowledge and the Netflix prize dataset as an anonymized dataset, and were indeed able to recognizes users. Furthermore, the two researchers applied a similar technique to social networks and showed that the network topology in these networks can be used to re-identify users [41]. Recently, Wondracek et al. [42] introduced a novel technique based on social network groups as well as some traditional browser history-stealing tactics to reveal the actual identity of users. They based their empirical measurements on the XING network, and their analysis suggested that about 42% of the users that use groups can be uniquely identified.

One of the prerequisites for being able to launch the automated attacks is the ability to break CAPTCHAs used by a site. Several projects in the area of computer vision exist that provide libraries to break real-world CAPTCHAs (e.g., [43, 44]). Note that our main focus is not to advance the field of CAPTCHA breaking, but to be able to break the CAPTCHAs efficiently enough to be able to automate the attacks that we describe. Obviously, some CAPTCHAs are easier to break than others (e.g., StudiVZ and XING are simpler than the reCAPTCHAs employed by Facebook).

13.5 Conclusions

Social networking sites have been increasingly gaining popularity. Many social networking sites have millions of registered users now. Unfortunately, when a new technology starts to attract a large number of Internet users, criminals are attracted as well. Today, it is not uncommon for Internet users to be participants in more than one social networking site (e.g., LinkedIn for business, and Facebook for private networks).

In this chapter, we gave a summary of two types of attacks we have presented in the past [18, 19]. The reader is referred to these papers for more details. The first attack involves the profile cloning of existing user accounts either to the same or to another social networking site. The second attack automatically exploits a common weakness that is present in many popular social networking sites. We are able to correlate collected user information across many different social networks. That is, users that are registered on multiple social networking web sites with the same e-mail address are vulnerable. For example, it is possible to identify users who are potentially using a different name on a dating website, and are pretending to be younger than they really are. The correlation that we are able to do automatically has a significant privacy impact.

References

1. Berkowitz, S.D.: An introduction to structural analysis: The Network Approach to Social Research. Butterworth, Toronto, ISBN 0409813621 (1982)
2. Xing—Global networking for professionals. http://www.xing.com (2008)
3. LinkedIn. http://www.linkedin.com (2008)
4. Facebook. http://www.facebook.com (2008)
5. MySpace. http://www.myspace.com (2008)
6. StudiVerzeichnis—StudVZ. http://www.studivz.net (2008)
7. MeinVerzeichnis—MeinVZ. http://www.meinvz.net/ (2008)
8. Facebook by the numbers. http://www.fastcompany.com/magazine/115/open_features-hacker-dropout-ceo-facebook-numbers.html (2008)
9. The spamhaus project. http://www.spamhaus.org/ (2008)
10. New myspace and facebook worm target social networks. http://www.darknet.org.uk/2008/08/new-myspace-and-facebook-worm-target- social-networks (2008)
11. CERT advisory CA-2000-04 love letter worm. http://www.cert.org/advisories/CA-2000-04.html (2008)
12. Spear phishing: highly targeted phishing scams. http://www.microsoft.com/protect/yourself/phishing/spear.mspx (2006)
13. Modeling and preventing phishing attacks. http://www.informatics.indiana.edu/markus/papers/phishing_jakobsson.pdf (2005)
14. Karlberger, C., Bayler, G., Kruegel, C., Kirda, E.: Exploiting redundancy in natural language to penetrate Bayesian spam filters. In: First USENIX Workshop on Offensive Technologies (WOOT '07), Boston, MA, August (2007)
15. Dwyer, C., Hiltz, S.: Trust and privacy concern within social networking sites: a comparison of facebook and myspace. In: Proceedings of the 13th Americas Conference on Information Systems (AMCIS) (2007)

16. Fogel, J., Nehmad, E.: Internet social network communities: Risk taking, trust, and privacy concerns. Comput. Hum. Behav.**25**(1), 153–160 (2009)
17. Gross, R., Acquisti, A.: Information revelation and privacy in online social networks. In: ACM Workshop on Privacy in the Electronic Society (WPES) (2005)
18. Bilge, L., Strufe, T., Balzarotti, D., Kirda, E.: All your contacts are belong to us: automated identity theft attacks on social networks. In: 18th International Conference on World Wide Web (WWW) (2009)
19. Balduzzi, M., Platzer, C., Holz, T., Kirda, E., Balzarotti, D., Kruegel, C.: Abusing social networks for automated user profiling. In: nternational Symposium on Recent Advances in Intrusion Detection (RAID) (2010)
20. Jagatic, T.N., Johnson, N.A., Jakobsson, M., Menczer, F.: Social phishing. Commun. ACM **50**(10), 94–100 (2007)
21. Brown, G., Howe, T., Ihbe, M., Prakash, A., Borders, K.: Social networks and context-aware spam. In: ACM Conference on Computer Supported Cooperative Work (CSCW) (2008)
22. News, H.: Spam-Bots werten soziale Netze aus http://www.heise.de/security/Spam-Bots-werten-soziale-Netze-aus–/news/meldung/145344, September 2009
23. Douceur, J.R.: The sybil attack. In: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), March (2002)
24. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: SybilGuard: defending against sybil attacks via social networks. The Proceedings of ACM SIGCOMM '06 (2006)
25. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: SybilLimit: a near-optimal social network defense against sybil attacks. In: IEEE Symposium on Security and Privacy (2008)
26. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Gossip algorithms: Design, analysis and applications. In: IEEE INFOCOM (2005)
27. Flaxman, A.D.: Expansion and lack thereof in randomly perturbed graphs. Internet Mathematics 4(2) (2007)
28. Sophos facebook ID probe. http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html (2008)
29. Bonneau, J., Preibusch, S.: The privacy jungle: on the market for privacy in social networks. In: Workshop on the Economics of Information Security (WEIS) (2009)
30. Chew, M., Balfanz, D., Laurie, B.: (Under)mining privacy in social networks. In: Proceedings of Web 2.0 Security and Privacy Workshop (W2SP) (2008)
31. Jones, S., Millermaier, S., Goya-Martinez, M., Schuler, J.: Whose space is MySpace? A content analysis of MySpace profiles. First Monday, **12**(9), August (2008)
32. Krishnamurthy, B., Wills, C.E.: Characterizing privacy in online social networks. In: Workshop on Online Social Networks (WOSN) (2008)
33. Bonneau, J., Anderson, J., Danezis, G.: Prying data out of a social network. In: First International Conference on Advances in Social Networks Analysis and Mining (2009)
34. Chau, D.H., Pandit, S., Wang, S., Faloutsos, C.: Parallel crawling for online social networks. In: 16th International Conference on World Wide Web (WWW) (2007)
35. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: ACM SIGCOMM Conference on Internet Measurement (IMC) (2007)
36. Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y.: User interactions in social networks and their implications. In: 4th ACM European Conference on Computer Systems (EuroSys) ACM (2009)
37. Griffith, V., Jakobsson, M.: Messin' with texas, deriving mother's maiden names using public records. In: Third Conference on Applied Cryptography and Network Security (ACNS), June 2005
38. Raymond Heatherly, M.K., Thuraisingham, B.: Preventing private information inference attacks on social networks. Technical Report UTDCS-03-09, University of Texas at Dallas (2009)

39. Irani, D., Webb, S., Li, K., Pu, C.: Large online social footprints–an emerging threat. In: IEEE International Conference on Computational Science and Engineering, **3**, 271–276 (2009)
40. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: IEEE Symposium on Security and Privacy (2008)
41. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: EEE Symposium on Security and Privacy (2009)
42. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A practical attack to de-anonymize social network users. In: EEE Symposium on Security and Privacy (2010)
43. kloover.com. Breaking the ASP Security Image Generator. http://www.kloover.com/2008/02/28/breaking-the-asp-security-image-generator/
44. PWNtcha. PWNtcha—captcha decoder. http://sam.zoy.org/pwntcha/

Chapter 14
Decentralized Social Networking Services

Thomas Paul, Sonja Buchegger and Thorsten Strufe

Abstract Online Social Networks (OSN) of today represent centralized repositories of personally identifiable information (PII) of their users. Considering their impressive growth they arguably are the most popular service on the Internet, both by technology savvy but even more by comparably inexpert audiences, today. Being voluntarily maintained and automatically exploitable, they are a promising and challenging target for commercial exploitation and abuse by miscreants. Several approaches have been proposed to mitigate this threat by design. Removing the centralized storage, they distribute the service and data storage, to protect their users from a provider that has access to all the information users put into the system. This chapter gives an overview of currently proposed approaches, and classifies them according to their core design decisions.

Keywords Online Social Networks · Security · Peer-to-Peer · Decentralization

14.1 Introduction

Online Social Networks (OSN) are currently revolutionizing the way people interact, and are becoming de facto a predominant[1] service on the web. The impact of this paradigm change on socio-economic and technical aspects of collaboration

[1] http://blog.nielsen.com/nielsenwire/online_mobile/top-u-s-web-sites-and-brands-for-april-2010/

T. Paul · T. Strufe (✉)
University of Mannheim, Mannheim, Germany
e-mail: strufe@ieee.org

T. Paul
e-mail: thomas.paul@cs.tu-darmstadt.de

S. Buchegger
e-mail: buc@csc.kth.se

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_14,

and interaction is comparable to that caused by the deployment of the World Wide Web in the 1990s.

Catering to a broad range of users of all ages and a vast difference in social, educational and national background, OSN allow even users with limited technical skills to publish Personally Identifiable Information (PII) and to communicate with ease, sharing interests and activities with their friends or indeed anybody on the web. Online Social Networks contain digital representations of a subset of the relations that their users, both registered persons and institutions, cultivate in the physical world. Centralized Social Network Services (SNS) manage, and offer online access to these OSN.

Adapted from the original definition in [4], an online social network can be defined as an online platform that (1) provides services for a user to build a public profile and to explicitly declare the connection between his or her profile with those of the other users; (2) enables a user to share information and content with the chosen users or public; and (3) supports the development and usage of social applications with which the user can interact and collaborate with both friends and strangers.

In centralized OSN all personal content is stored, at least logically, at a single location. This data store contains a very valuable collection of private information. Centralized OSN need an operator to provide for the resources and to maintain this service. Their primary way of financing is based on advertising, and since the detailed personal information about clients and potential new customers is very useful for the advertising industry, this property is vital. The idea to utilize this data treasure to gain money for the Social Network Provider is not far. The value of this exposed, private data is underlined by the market capitalization of OSN providers, which ranges from 580 million US$ (acquisition of myspace through the news corp. in 2005) to 23 billion US$ (Facebook Inc, according to the investment of Elevation Partners in 2010).[2] Even when one considers the commercial bodies that provide SNS to be trusted entities, hackers may be able to compromise their systems to gain access. Unsatisfied employees may abuse their access to the data,[3] or even imprudent publication of seemingly anonymized data may lead to the disclosure of PII, as it has happened in the past.[4] In consequence, the protection of the data published in OSN is an emerging, important topic. It is currently not satisfyingly addressed by the providers in an appropriate way, a deficit that is not very likely to change.

Responding to these undesirable properties, several different solutions have been proposed to provide SNS features while preserving the users' privacy. These range from the application of simple dictionaries [11], more sophisticated cryptographic methods [2], to the distribution of the data store [6, 8, 17] to avoid a single, omniscient provider. Decentralizing the social networking services

[2] http://www.reuters.com/article/idUSTRE65S0CZ20100629

[3] http://www.techcrunch.com/2010/09/14/google-engineer-spying-fired/

[4] http://www.nytimes.com/2006/08/09/technology/09aol.html

promises a comprehensive privacy preservation of the users' data and shall be analyzed further in the remainder of this chapter.

14.2 Functional Overview of Online Social Networks

Social networking sites developed from early, simple online tools to manage personal and professional contacts to effective services for sharing various information and content. Popular OSN, such as, e.g., Facebook, offer users even more functionality and applications, as third-parties are allowed to develop and plug their applications into the site. Online Social Networks in this course have come closer to being full-fledged development and management platforms for social applications.

Even though each OSN is usually tailored to some specific use, like, e.g., Xing[5] or LinkedIn[6] for professional contacts and MySpace[7] or MeinVZ[8] for private friends, the functional range of these platforms is essentially quite similar. Generally speaking, OSN functionality can be classified into two main types, networking and data functions. The networking functions serve the actual purpose of the OSN to foster social relationships amongst users within the virtual platform. In particular, they provide functionality for building and maintaining the social network graph. The data functions are responsible for the management of user-provided content and communication amongst the users. Their variety contributes to the enhancement of users' interaction and makes the platform more attractive.

14.2.1 Networking Functions

Online Social Networks users may build their profiles and establish relationships with each other. The set of networking functions includes all functions that update the vertices and the edges of the social network graph. In particular, the OSN user invokes the profile creation function upon his or her registration on the OSN platform. This function adds a new vertex representing that user to the social network graph. Thereafter, with profile lookup the user can find other users, who are also represented via vertices. Through the call to the relationship link establishment function the user can set up a new relationship with some other user. This function sends notification to that user, who in turn can accept or ignore the request. If the user accepts the request then users are added to the contact lists of

[5] http://www.xing.com/de/
[6] http://www.linkedin.com/
[7] http://www.myspace.com/
[8] http://www.meinvz.net/

each other and a new edge representing their relationship is added to the social network graph. The OSN users can also encounter profiles for possible relationships by browsing through the contact list function, which is realized through the traversal along the edges of the graph. Additional networking functions can be used to remove vertices and edges from the graph, for example upon the deletion of a user's profile.

14.2.2 Data Functions

Online Social Network users can typically advertise themselves via their own profiles and communicate with each other using various applications like blogs, forums, polls, chats, e-mails, and online galleries. Here we point out the profile update function, which allows the OSN users to maintain details on their own profiles and provide fresh information to other users, who may call the profile retrieval function, and hence visit the profile. Communication amongst users via blogs and forums is typically implemented through a posting function, which inserts a block of information as an element into the main thread (sometimes called the "wall"). This block of information is not limited to plain text and can also contain videos, pictures, or hyperlinks. Updates to the profile or main thread, or a subset thereof, often are shown as a news feed on the main OSN page of connected users.

An OSN user willing to set up multimedia galleries typically calls the upload function, which transfers digital data from the user's device to the OSN database. In case of content depicting other users, the tag function can be used to create a link pointing to their respective profile. Online Social Networks users can typically evaluate content published by other users through the like or dislike functions. Using the comment function OSN users can articulate their point of view in a more explicit way. Online Social Networks users can also exchange personal messages. Here, in particular, basic asynchronous offline communication (comparable to email) is implemented, and synchronous real-time communication between online users is offered in the form of chats. Online Social Network users can send messages to individuals or to subgroups of users from their contact list. Additionally, users may create interest groups.

14.3 Decentralized Online Social Networks

All users of centralized OSN, who request the service, cause traffic at the social networking service provider. Growing to several millions of users, these central services naturally become bottlenecks. This problem has become increasingly apparent with the frequent down times and service break downs that users of

highly popular OSN have experienced in the recent past.[9] Decentralized OSN avoid this disadvantage by distributing and making the stored data available from multiple locations. This inherently leads to a protection of the data from unintended centralized access and exploitation.

Decentralizing the service provision, however, add some new challenges [5] and ***requirements***. The distribution of OSN, which are catering for a broad range of users that frequently include large inexpert audiences, needs to be entirely transparent. Access to the data and functions needs to be provided through a single *integrating interface* which has to allow for easy publication, search, and retrieval of profiles and attributes. All *data related functions* of centralized OSN have to be provided in addition. The possibility to *reconstruct the social graph* of relations between the users finally has to be provided as well, in order to allow for simplified, often publish-subscribe-like communication, ease of access control, and publicly announcing real world friend- and other relationships. The distribution additionally must not lead to an interrupted *availability of data and services*, not even of parts thereof, even in face of the transition from dedicated servers to distributed resources. *Confidentiality* has to be met and *access* to each attribute *controlled*, even considering the lack of centralized management and control. Preserving the *privacy* of users and their data, even to an extent at which they are able to hide their participation inside the OSN completely, needs to be supported.

Online Social Networks in general may be decentralized at a different granularity. Integrating multiple commercial OSN [3, 12] and keeping chosen, partial data within the bounds of different SNS represents a simple step towards decentralization. This approach removes the omniscient, commercial service provider with access to the overall set of PII of the users. It however introduces the role of the aggregator, which, even though only catering for a subset of all participants in the integrated OSN, again can collect complete knowledge about its users. To achieve better decentralization, the service provision can further be distributed, up to a granularity of a single service provider for each user. We can distinguish two groups of decentralization: web-based and peer-to-peer.

14.3.1 Web-based Decentralized Online Social Networks

Systems in this first group (mainly comprising of diaspora[10,11] and "Friend-of-a-Friend" (FoaF) [17]) leverage on a distributed web server infrastructure. They require the acquisition of webspace or the deployment of additional web servers

[9] http://www.allfacebook.com/2009/08/facebook-downtime-issues/
http://www.pcworld.com/article/173550/facebook_outage_silences_150000_users.
http://www.pingdom.com/reports/vb1395a6sww3/check_overview/?name=twitter.com%2Fhome

[10] http://www.joindiaspora.com

[11] diaspora still is in the course of development and subject to rather rapid design changes. It is considered as it has been proposed during the time of this writing.

through their participants. Users then can publish their profiles much alike web pages in their own web space and locally manage access control rules to specifically allow retrieval of restricted attributes and resources to selected users. Web-like links to the profiles of other users are employed to represent the contacts list, and hence recreate the social graph.

The main *challenge* for these systems is their need for access to reliable web space, without which the profiles of the respective users are unavailable. Many, especially less tech savvy users experience major difficulties when being confronted with the task of setting up a web server themselves. Especially the task of reliably providing this service from home, including the configuration of home gateways, NAT, and firewalls, represents a serious obstacle. Renting web space, on the other hand either comes with the lack of being able to implement fine grained access control, and is quite costly in comparison to the existing free OSN, or does not help decreasing the complexity—and difficulties—of administrating them. This challenge of course generates a new business model, the provision of reliable, pre-taylored web space, including massive data aggregation at the provider and the resulting adverse consequences for the privacy of its users. A systematic challenge to these systems is the possibility to search for profiles of other users, like the proverbial long-lost-friend, since this is difficult to be implemented inside the systems. It rather has to be implemented in search engines, which again can gather knowledge about the users at a large extent.

14.3.2 P2P Online Social Networks

The second group of systems [1, 6, 8, 10] harnesses the advantages of the well-known Peer-to-Peer principle [16] in order to allow for the publication, search, and retrieval of profiles and their attributes, much alike the sharing, searching, and downloading in conventional P2P filesharing systems.

Challenges for P2P OSN are mainly caused by the different properties of file sharing vs. social networking. P2P file sharing systems have been designed for the purpose of reliably distributing comparably few, large, popular data objects (music files, movies). The automatic replication of these files during download led to an inherent load balancing, since more popular resources are downloaded, and hence replicated, more often. File sharing systems, however, only offer best-effort services, and the availability of less popular resources is all but guaranteed. Considering profiles in OSN exhibits a drastically different situation. Data in OSN consist of a profile for each user, each of which comprising a plethora of personal attributes. All these attributes of the large majority of users enjoy a very low popularity, but even though requested only very occasionally have to be kept available at all times. Owner replication, the provision of data at downloading parties, requires some registration of each resource (for the purpose of finding a replica), which becomes a difficult task considering the sheer numbers of single attributes. Another adverse property along the same lines is the fact that while the

Table 14.1 Classification and properties of the analyzed systems

Approaches	Type of storage	Storage granularity	Level of Integration	Resource Sharing
diaspora	Web-based	Complete	External services	Premium services
FoaF	Web-based	Complete	External services	
LifeSocial	Hybrid	Split	Stand alone	Premium services
Likir	P2P	Split	Stand alone	
PeerSoN	P2P	Split	External services	
Safebook	P2P	Complete	Stand alone	Cooperation
Vis-a-Vis	Dedicated	Complete	External services	

data in file sharing generally is accessible by anyone, access to private attributes of the profiles is restricted and they hence may not be replicated at arbitrary peers. Timing constraints are difficult to meet in P2P OSN: While users of file sharing systems are willing to wait even comparably long intervals to download a complete movie or song, the user of an online social network expects the requested profile to be represented with very low delays. Even the user behavior poses a significant challenge for P2P OSN: the users of file sharing systems are willing to stay online as long as it takes to download rather large files, whereas the users in P2P OSN will usually login, browse a few profiles, send a few messages, and log out after having been online for a couple of minutes, only; providing a reliable peer-to-peer data service in this scenario causes serious obstacles for the system designers.

Peer-to-Peer OSN in conclusion also face serious challenges when striving at providing reliable social networking services.

14.4 Classifying Decentralized Social Networking Services

Analysing decentralized social networking services described in 14.3.1 and 14.3.2, they can be classified according to a few distinguishing characteristics. Their main target being the publication of profile information, while preserving the privacy of their users, they follow central design choices according to the four following properties:

1. The type of storage
2. The granularity of storage
3. The level of integration
4. Resource sharing incentives.

The properties and classified groups are further explained in the following sections. Table 14.1 gives an overview of the analyzed systems and their classification.

14.4.1 Type of Storage

Depending on the type of storage, the approaches can be classified into two groups. The first group, mainly consisting of the web-based approaches (diaspora and FoaF [17]) as well as Vis-a-Vis [15] leverage on dedicated servers. FoaF and diaspora on the one hand assume access to dedicated web space at which the profiles of users can be stored and retrieved. Vis-a-Vis on the other hand proposes to replicate the complete P2P software to a virtualized server in the cloud. Dedicated services, of course, come at an explicit, additional cost.

Likir [1], PeerSoN [6] and Safebook [8] propose to use local and shared resources of the P2P overlay. Leveraging on the rather unreliable storage services of other peers, who are subject to churn themselves, requires more sophisticated means of keeping the data available, which in turn causes a higher overhead and implicit cost, shared between the participants of the system.

LifeSocial [10] represents a hybrid approach. It implements a PAST [9] reliable P2P storage between the participating nodes, and additionally allows to acquire storage space at a dedicated server as premium services, for the purpose of guaranteeing the availability of data.

14.4.2 Granularity of Storage

The granularity of remote storage ranges from replicating the complete service at the same place to storing each attribute at different places in the system.

The web-based approaches (diaspora, FoaF [17]), as well as Vis-a-Vis [15] and Safebook [8], bundle the complete service of delivering profiles. While the web-based approaches place the whole profile remotely in a single web space, Vis-a-Vis migrates the P2P software to a virtual server entirely. Safebook creates multiple replicas of the complete profile, one at each of the profile owner's direct friends.

The remaining approaches (Likir [1], LifeSocial [10], and PeerSoN [6]) split the profile, which might be quite large in volume, into its attributes and also allow to replicate each attribute at different places. The load of storing data for others consequently may be balanced more evenly, which in turn may lead to a reduced need for incentives for cooperation. This can, however, cause increased messaging overhead for the location and retrieval of each of the attributes, and eventually the complete profile.

14.4.3 Level of Integration

Implementations of SNS may either be self-contained, or integrating other services.

One group of approaches can be considered fully-fledged, stand-alone SNS, completely integrating the functionality and providing means to keeping data available anytime. These especially comprise of Likir [1], LifeSocial [10], and Safebook [8].

The other group of systems leverages external services for the replication and availability guarantees. Vis-a-vis [15] envisions to replicate the complete service to the cloud, which is expected to offer reliable availability, when the user is offline.

The first prototype implementation of PeerSoN [6] uses a third party DHT (openDHT [14]) as a lookup service to find content, and the SNS peers for storage. OpenDHT can be replaced by any DHT offering similar service (put, get, remove of entries) or by a self-contained peer implementation, combining the functionalities of storage and information administration in one system. PeerSoN also envisions the option of using dedicated services that have high online probabilities, such as home routers or individual cloud storage, for users whose mobile or desktop resources are limited in extent and availability. Leveraging on external services, while potentially increasing the availability of data, comes at the cost of depending on them. Break downs and performance deficiencies have a direct impact on the operation of the SNS. Integration of commercial services, like cloud storage or computing most certainly cause additional cost.

These classes do not identify web-based decentralized SNS (diaspora, FoaF [17]) very well, since they are integrated into the web, rather than being stand-alone systems. They hence mainly comprise of a web page description scheme, and possibly an interface for their usage.

14.4.4 Resource Sharing Incentives

Implementing an integrated SNS with replication that cannot rely on external storage systems results in the need to incentivize service providers to actually store the replicas, to keep them available, and to eventually deliver them to requesting users.

Different incentive schemes have been proposed in literature that could potentially be integrated and simply utilized. However, none of the existing approaches follows this strategy. While not all of the proposed approaches actually consider this need, the chosen solutions can be generalized to two different types: financial and social incentives.

Some solutions, like, e.g., diaspora, or LifeSocial [10], consider the possibility of offering payed premium services through the system provider, which hence enjoys *financal incentives*. These premium services would comprise a centralized replication of the premium profiles for a fee, in order to keep them available at all times.

Safebook [8] takes another approach of considering *social incentives*: since friends of a user generally are trusted and believed to cooperate, the main profile

information of each user is replicated to all their friends' devices. Complex, additional networked structures, the "matryoshkas", are created for the purpose of hiding the friend relationship from other participants, and they are optimized to increase the chance of locating the profile replicas. However, the availability of profiles may not be guaranteed, if the number of friends of a user is too low, or in case that all of a user's friends concurrently are offline.

14.5 Alternatives to Decentralization

The benefit of decentralization of social networks is, as shown before, the potential for increased privacy and reliability. But decentralization is not the sole approach targeting these goals. "None of your business" (NOYB) [11] proposes to share dictionaries between trusted users. Profile owners substitute their true attributes according to the shared attributes and participants with knowledge of these dictionaries can replace the seemingly useless data with the initial, proper information. Persona [2] employs more sophisticated cryptographic methods and uses attribute-based encryption which allows users to build fine-grained access control. Users of a social network with central infrastructure, based on Persona, do not need to trust their provider as much as it is necessary in unencrypted environments. However, some information about the using habits, the contact list and the simple fact that a user is a member of this social network remains accessible. This concept does also not address the issue of the bottleneck and the single point of failure in the infrastructure as well as economical interest of a service provider to gain money for managing the maintenace costs.

An important functionality of social networks is the ability to communicate with people in the network, be it synchronous or asynchronus. The eXtensible Messaging and Presence Protocol[12] (XMPP) is a communications protocol, based on XML, which supports Instant Messaging (IM) and the management of contact lists. Everybody in possession of a DNS domain may deploy their own server, and leveraging the Domain Name System these distributed servers are enabled to locate and establish connections between each other. Using the distributed XMPP servers and user IDs assigned to their hosting domain names, XMPP allows for decentralized communication. Considering those similarities and the issue of decentralized communication infrastructure in using XMPP, several approaches have been proposed to enrich XMPP for building decentralized OSN. Numerous projects, like, e.g., Jappix[13] and onesocialweb,[14] exist. However, since they do not offer the full functional range of OSN, frequently lacking the possibility to search

[12] http://xmpp.org/

[13] http://www.jappix.com/

[14] http://onesocialweb.org/

for other users and to easily browse their social network, they have been omitted in this analysis.

A different, yet closely related group of systems are darknets. Approaches like freenet [7], turtle [13], or WASTE[15] aim at providing comprehensive anonymity between the users of file sharing systems. They leverage on social trust and each participant's device only connects to the devices of real-life acquaintances, much alike the approach of Safebook. However, their focus is the anonymous publication of data rather than allowing a low-delay access to it, and they hence are not applicable to provide general purpose OSN.

14.6 Conclusions

The information we reveal about ourselves online has changed both in quantity and, as it is becoming increasingly personal, in quality, over the last decade. In parallel, web services based on an advertising business model have gained market share. Both trends are well exemplified by online social networks. The model of an attention economy that such business models (based on advertising) are building on, is by necessity one of scarcity. Our attention as humans is limited by the hours in a day. Given this rather hard limit on how many effective advertisement-based services can co-exist, advertisement has to be targeted to user interests in order to get enough click-through and impressions to support the service. The more targeted the advertising, the more personal and demographical information about prospective customers is needed. This model thus renders information about users more valuable, resulting in an incentive for service providers to gather even more personal information.

We therefore increasingly observe that service providers, especially those of SNS, are pushing the boundaries of extracting personal information from users. Online social network providers have been attracting users to share an increasing range of personal data that is shared along the lines of friends, friends of friends, other users, and, finally, anybody on the Internet. Despite outcries about privacy violations and difficulties of configuring privacy setting preferences, this trend continues. Following pressure from users, Facebook, for example, recently changed the way privacy settings are made in an effort to make it easier for users. Their default settings, though, are quite far from what one would expect as privacy preserving. The typical user who relies on default settings thus gets privacy settings that share vital personal information with everyone on the Internet.

In an effort to preserve user privacy while keeping useful features offered by online services, such as social networks, there is increasing research activity proposing to go from centralized provider-based models toward a community-driven decentralized approach, usually based on the well understood peer-to-peer

[15] http://slackerbitch.free.fr/waste/

principle. In this chapter, we discussed several of these approaches and classified them according to design decisions such as whether they are web- or P2P based, whether they integrate third-party services, how storage is provided, and others. This is a current snapshot of research projects for decentralized SNS and we anticipate more and different approaches in the near future. Our survey and classification serves as a first step toward distilling best practices from different approaches to decentralizing SNS. Over time, given lessons learnt from implementations, experiments, and hopefully even user adoption, such classifications and evaluations enable designers of decentralized SNS to leverage results from others and build privacy-preserving SNS that exhibit desirable features such as low overhead, high availability, and reliability.

References

1. Aiello, L.M., Ruffo, G.: Secure and flexible framework for decentralized social network services. In: IEEE International Workshop on Security and Social Networking (SESOC) (2010)
2. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. In: SIGCOMM (2009)
3. Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: ACM Internet Measurement Conference (2009)
4. Boyd, D.M., Ellison, N.B.: Social network sites: definition, history, and scholarship. J. Comput. Mediat. Commun. **13**(1), (2007) article 11
5. Buchegger, S., Datta, A.: A case for P2P infrastructure for social networks—opportunities and challenges. In: Proceedings of WONS 2009, The Sixth International Conference on Wireless On-demand Network Systems and Services. Snowbird, USA (2009)
6. Buchegger, S., Schiöberg, D., Vu, L.H., Datta, A.: PeerSoN: P2P social networking—early experiences and insights. In: Workshop on Social Network Systems (2009)
7. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Workshop on Design Issues in Anonymity and Unobservability, pp. 46–66 (2000)
8. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: a privacy preserving online social network leveraging on real-life trust. IEEE Commun. Mag. **47**(12), 94–101 (2009)
9. Druschel, P., Rowstron, A.: Past: a large-scale, persistent peer-to-peer storage utility. In: ACM symposium on Operating Systems Principles (SOSP) (2001)
10. Graffi, K., Podrajanski, S., Mukherjee, P., Kovacevic, A., Steinmetz, R.: A distributed platform for multimedia communities. In: International Symposium on Multimedia (2008)
11. Guha, S., Tang, K., Francis, P.: NOYB: Privacy in online social networks. In: First Workshop on Online Social Networks (2008)
12. Guy, I., Jacovi, M., Shahar, E., Meshulam, N., Soroka, V., Farrell, S.: Harvesting with sonar: the value of aggregating social network information. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 1017–1026 (2008)
13. Matejka, P.: Security in peer-to-peer networks. Master's thesis, Charles University, Prague (2004)
14. Rhea, S., Godfrey, B., Karp, B., Kubiatowicz, J., Ratnasamy, S., Shenker, S., Stoica, I., Yu, H.: OpenDHT: a public DHT service and its uses. In: SIGCOMM (2005)
15. Shakimov, A., Cox, L., Varshavsky, A., Caceres, R.: Privacy, cost, and availability trade-offs in decentralized osns. In: Workshop of Online Social Networks (2009)

16. Steinmetz, R., Wehrle, K. (eds.) Peer-to-Peer Systems and Applications. Lecture Notes in Computer Science, vol. 3485. Springer, Heidelberg (2005)
17. Yeung, C.M.A., Liccardi, I., Lu, K., Seneviratne, O., Berners-Lee, T.: Decentralization: the future of online social networking. In: W3C Workshop on the Future of Social Networking Position Papers (2009)

Part IV
Secure Collaborative Systems

Chapter 15
Access Control, Privacy and Trust in On-line Social Networks: Issues and Solutions

Elena Ferrari

Abstract On-line Social Networks (OSNs) are today the hugest repository of personal information available on the Web. Such great amount of personal information gives us a unique opportunity in that the possibility to trace and analyze complex dynamic networks describing the evolution of relationships (among individuals or organizations) could change the way we understand complex phenomena such as economic/financial ones, social trends, fashions, opinions, interests, or the generation and dissemination of consensus and trustworthiness. For instance, we can understand how individual behaviours, i.e., small choices at a local level, can cause global effects. On the other hand, the availability of this huge amount of information poses new challenges in terms of access control and privacy protection. The aim of this chapter is to briefly review the state of the art with respect to the protection of resources shared in an OSN and to highlight some of the most promising research trends in the area.

Keywords On-line social networks · Access control · Privacy · Trust

15.1 Introduction

On-line Social Networks (OSNs) represent one of the most relevant phenomena related to Web 2.0. OSNs are on-line communities that allow users to publish resources and record and/or establish relationships with other users, possibly of different type ("friend of", "colleague of", etc.), for purposes that may concern business, entertainment, religion, dating, etc. To have an idea of the relevance of the social networking phenomenom, just think that today the most widely used

E. Ferrari (✉)
DICOM, University of Insubria, Varese, Italy
e-mail: elena.ferrari@uninsubria.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_15, © Springer-Verlag Italia Srl 2011

OSN (i.e., Facebook—www.facebook.com) claims to have more than 500 million active users. These rapid widespread of social network facilities has resulted in the fact that today OSNs are becoming the hugest repository of available personal information. This poses both interesting opportunities and challenges in particularly in the field of privacy and access control [4].

Up to now the research in the field of access control and privacy in OSNs has mainly focused on *off-line privacy*, that is, privacy-preserving analysis or publishing of social network data [3]. However, privacy is not a primary concern only when social network data are analyzed off-line, but also during the normal activities of users within an OSN, as also witnessed by the increasing attention to this issue from media, privacy advocates, and citizens (see for instance [11]). However, there are also people that have a completely different view with respect to privacy/confidentiality protection in OSNs. Recently, Facebook founder Mark Zuckerberg in an interview with the weblog TechCrunch argued that "Privacy is no longer a social norm, because people have really gotten comfortable not only sharing more information, but more openly and with more people. That social norm is just something that has evolved over time". Clearly, many people do not agree with Zuckerberg's view, but it should be recognized that concepts such as privacy and confidentiality have evolved over time and that accessible social networking technologies have greatly contributed to this evolution. Therefore, we need new methods and a completely different view in addressing privacy/confidentiality issues in OSNs, able to go beyond traditional mechanisms. Indeed, in a traditional Data Management System the user population is known in advance (e.g., the employees working in an organization). This is not true for OSNs. Therefore, we believe that central to the challenge of protecting data in an OSN is the concept of *trust*. Trust [10] is a relationship between a pair of users. Generally speaking, the propagation of information, especially when it involves private data, should be driven by the confidence residing between two persons, namely the existence of different level of trust or distrust. The key question is therefore how trust can be measured in an open and dynamic environment like an OSN.

This chapter will try to shed light into the research challenges in the area of access control, privacy and trust in OSNs, by briefly reviewing the state of the art and discussing some of the main new issues related to the protection of OSN user personal data and resources. More precisely, the remainder of this chapter is organized as follows. Next section briefly review the state of the art, whereas Sect. 15.3 discusses some of the most promising research directions in the field, that is, those related to privacy-aware access control, trust modelling and computation, and risk analysis. Finally, Sect. 15.4 concludes the chapter.

15.2 State of the Art

In what follows, we briefly review the state of the art by focusing on the enforcement of privacy and confidentiality requirements during the normal

activities of OSN users. We refer the reader to [3, 19] for privacy-preserving data publishing and mining of OSN data.

In the field of a controlled information sharing in OSNs, recently some research proposals have appeared, aiming to overcome the restrictions of the protection mechanisms provided by current OSNs. The majority of these proposals enforce *topology-based access control*, according to which confidentiality/privacy requirements are expressed by stating conditions on the social graph of OSN users. Such constraints are expressed in terms of relationship types, their depths and possibly their trust levels, depending on the considered model. One of the first proposal of a topology-based access control model is the one by Carminati et al. [9], where resource owner access control requirements are expressed in terms of a set of access rules associated with the resource to be protected. Access rules denote authorized members in terms of the type, depth and trust level of the relationships they must have with other network nodes in order to be authorized for the access. Access control is *requestor-based* in that the burden of access control is mainly on the requestor side that must provide the resource owner with a proof of being authorized to access the requested resource. Since access control is topology-based, the proof shows the existence of a path in the OSN with the characteristics (in terms of relationship type, depth and trust) required by the access rules associated with the requested resource. Relationship information are coded into certificates, signed by both the parties involved in the relationship. Certificates are managed by a certificate server which is in charge of path discovery upon request.

In contrast, the model in [15] represents authorizations by means of access control lists (ACLs) associated with the shared resources. Also in [15] access control is topology-based in that the ACLs may contain the identifiers of authorized users as well as the relationships a user must have in order to gain access to the corresponding resource. Then, similar to [9], relationship certificates are exploited to enforce access control.

The use of mandatory access control in OSNs has been explored by [1]. According to mandatory access control, both resources and subjects are assigned a security level. The security level associated with a resource is a measure of its sensitivity, whereas in the case of subjects is a measure of their trustworthiness. Therefore, in the model presented in [1] each user u has an associated reputation value $r(u)$, which is computed as the average of the trust ratings specified for u by other users in the OSNs. This is associated as security level to the user. Resources created by a user u are assigned a security level equal to τ, where τ is a reputation value in the range $[0, r(u)]$ that the user has selected as his/her security level when he/she logged in the social network. Users are then authorized to access only resources with a security level equal to or less than τ. In [24] the same authors propose a topology-based access control model, according to which access control rights are determined based on the distance between the owner and the requestor. More precisely, users can be classified into three adjacent zones on the basis of their distance from a resource owner. The size of the zones is set by the user. Users falling in the "acceptance" zone are those whose access requests will be immediately accepted; users falling in the "attestation" zone are those whose

access requests require a further evaluation to eventually gain access, whereas users falling in the "rejection" zone are those whose access requests will be immediately rejected since they are too far away from the resource owner. Therefore, confidentiality/privacy requirements are specified in terms of two distances, called trusted distances, delimiting the three zones.

In contrast, the work reported in [14] adopts a different point of view since it considers Facebook as the target scenario. More precisely, [14] starts with an analysis of the Facebook access control mechanism, with the aim of extending and enriching it. Besides access control policies, [14] identifies other three policy types, that is, search, traversal and communication policies. Search and traversal policies have been identified since Facebook allows one to look for new users in the social network by accessing some parts of users' profiles as well as the users' friend lists (this allows a search by traversing the social graph). As Facebook makes users able to state privacy settings regulating the access to this information, these are also supported by the model proposed in [14]. The last policy type, called communication policy, aims to make users able to state who is authorized to initiate a given type of communication with him/her.

Other work in the field of access control for OSNs are those exploiting semantic Web technologies [21] to provide much richer modelling of social network data (e.g., representing relationships among users, resources and actions in details), on which more flexible and expressive protection mechanisms can be devised. For instance, besides ownership there are other relationships among users and resources that can be exploited for access control, e.g., a user can own a photo but can also be tagged in a photo and, because of this, he/she may take part in the related access control decisions. Along this line is the proposal in [13] where a semantic framework based on OWL—Web Ontology Language—has been presented for defining different access rights exploiting the relationships between the individuals and the community. In contrast, [7] proposes an extension of the topology-based access control model in [9], based on Semantic Web Rule Language (SWRL) [18]. In particular, social network related information (that is, user's profiles, relationships among users, relationships between users and resources, and actions) are encoded by means of an ontology. Access control policies are then specified as SWRL rules based on this ontology and therefore any reasoner that supports SWRL can be used to enforce access control. More details on the prototype implementation can be found in [7].

15.3 Research Challenges

In what follows, we discuss some of the most relevant research challenges in the field of access control, privacy and trust in OSNs.

15.3.1 Privacy-Aware Topology-Based Access Control

The main purpose of an OSN is to establish relationships with other users and exploit such relationships for sharing resources. Therefore, the most natural way of expressing access control/privacy requirements is to pose constraints on the users social graph. In that case, the releasing of a resource is conditioned to the fact that the resource requestor has a relationship (either direct or indirect) of a specific type with other OSN member(s). However, relationships convey sensitive information in that a user may not want to reveal to other users that he/she holds a relationship of a given type with another user, or even if the relationship is public a user would like to keep the trust level confidential.

Therefore, an important issue is to preserve relationship privacy while performing access control. We call this *privacy-aware access control*. According to topology-based access control this means developing a technique for privacy-preserving discovery of paths in an OSN. Related to this issue is the problem of how access control is enforced and by what entities of the OSN. Traditionally, access control is enforced by a trusted reference monitor hosted by the data management server. The reference monitor intercepts each access request and, on the basis of the specified access control/privacy policies, determines whether the access can be partially or totally authorized, or it must be denied. Adopting this solution in an OSN implies to delegate to the OSN manager the role of reference monitor. We do not believe that this solution fits very well the OSN scenario, since it implies to totally delegate to the OSN manager the administration and enforcement of access control policies. This means that users should trust the manager with respect to the correct enforcement of their policies. However, some recent events (see for instance [11]) have increased the concerns of OSN users with respect to the centralized management of their data. Additionally, under this model, the OSN manager knows all the relationships in the network and their trust levels and, therefore, the privacy of OSN users is not preserved. Therefore, we believe that a decentralized access control solution [26] is the best option with respect to the privacy and confidentiality guarantees of OSN users. However, each access control solution to be effectively applied must consider also another important dimension, that is, the efficiency of access control. Since, access control in an OSN is topology-based, answering an access request may require to verify the existence of specific paths within an OSN. This task may be very difficult and time consuming in a fully decentralized solution. Additionally, relationship privacy should be preserved during access control. An essential requirement of access control enforcement is therefore to devise implementation strategies able to trade-off between security/privacy and efficiency/scalability.

To this purpose, a variety of solutions for privacy-preserving path discovery have been so far proposed, ranging from cryptographic ones [8] to the enforcement of collaborative protocols [6, 12, 20]; however, none of them addresses all the requirements related to privacy-preserving path discovery in OSNs. For instance, the solution presented in [8] suffers from the high overhead implied by key management.

The solution in [6] enforces path discovery through a collaboration driven by the privacy preferences associated with OSN relationships. Such preferences specify who is authorized to be aware of a relationship by stating which kind of relationships should link him/her to the user who has established the relationship. A collaboration request is sent to a user only if he/she satisfies the privacy preferences associated with the relationships in the path built so far. The drawback of this solution is that all the nodes taking part in the collaboration are aware of the characteristics of the path being discovered. Moreover, the protocol is not able to avoid that a malicious user sends the collaboration request, and the related path information, to a user which does not satisfy the associated privacy preferences. A collaborative protocol to path discovery in OSNs has been also proposed in [12], where homomorphic encryption is used to make the resource owner able to see only the aggregated trust value but not the trust values of each relationship in the path. However, the access control policy language in [12] does not support the maximum depth of a relationship as a parameter to perform access control. Therefore, policies such as "Only my friends and the friends of my friends" cannot be specified. Moreover, the solution is not fully decentralized since a trusted entity is introduced in the framework to be contacted only in case of conflicts among the users in the network (for instance, when a user suspects that another one contributed with a fake trust level to path discovery or has modified the trust value inserted by another user during the collaboration process). Additionally, one of the main drawbacks of the above-mentioned collaborative protocols is that they assume that nodes are always online to collaborate in the path discovery process, an assumption which is unrealistic in the context of OSNs.

A proposal addressing the issue of off-line nodes management is the one reported in [20], where two nodes not connected by a direct relationship can compute the depth of all the paths between them without referring to other nodes, by exchanging information (i.e., tokens) which is pre-computed at an earlier stage. Path discovery is performed by applying secure multi-party computation techniques to the set of tokens held by two nodes. However, the protocol is only able to deal with discovery of paths with a specific length, whereas relationship type and trust are not considered. In addition, the protocol could be inefficient for real OSNs, especially in the case of nodes with high connectivity (which is often the case of OSN nodes) or when path to be discovered have a large depth.

15.3.2 Trust-Based Information Sharing

Trust is a relationship between a pair of users that may impact on the actions performed in an OSN in many ways. In general, as it happens in real life, trust should be the key parameter according to which access to OSN resources should be regulated. However, it is well known that there does not exist a unique definition of trust, whose definition may vary depending on the context and for which purposes it is used [16]. For instance, in peer-to-peer systems the trustworthiness

of a given peer mainly depends on its reliability in providing a given service. When trust is used to enforce controlled information sharing in OSNs, a first fundamental issue is to define suitable models for representing trust. In this scenario, the semantics of trust should be mainly related to the compliance with the specified access control policies and privacy preferences. A user is trusted if he/she does not enact unauthorized flow of information in the OSN.

Another key point is how to compute trust. Indeed, it is quite evident that assigning a wrong trust value to a potential malicious user could imply unauthorized releasing of information or unauthorized disclosure of personal data. A possible solution is to apply the same rational adopted in the real world, where the trust value assigned to a person is estimated on the basis of his/her reputation, which can be assessed taking into account the person behavior with regards to all the other users in the OSN. This implies, first, to identify which kind of actions in an OSN are meaningful at determining user behaviors, when trust is used to enforce a controlled and privacy preserving information sharing. Such actions are mainly related to the release of resources/personal data to other users in the network. The second issue is how such actions can be monitored and used for trust computation. In this respect, a possible solution is the use of mechanisms based on audit files [22]. Audit files are then processed to verify the compliance of the actions they record to the specified access control/privacy policies. Audit files should be processed in a privacy-preserving manner, in such a way that an accurate measure of trust is obtained without leaking user personal information. However, alternative methods can be devised for trust computation which avoid to rely on a log file tracking user activities. For instance, a possibility that it is worth to be explored is to measure trust as a consequence of information propagation and/or the existence of specific *trust patterns* in the network. Pattern mining algorithms (e.g., [17]) can be adapted to this purpose. Another complementary option that can be investigated is to borrow some techniques from the field of trust negotiations [25]. According to the trust negotiation paradigm, trust between two initially untrusting parties is built as the result of an exchange of credentials between them. Since credentials may contain sensitive information, they are protected by (local to the parties) disclosure policies. Such approach has been successfully deployed in peer-to-peer architectures by managing trust relationships among different groups of peers [23]. In the context of OSNs, trust relationships between two users may be dynamically adjusted based on the outcome of negotiations involving the two users or some of their contacts. A preliminary work in this direction is the one reported in [5].

15.3.3 Risk Analysis Tools on Support of Access Control Policy and Privacy Preference Specification and Monitoring

One of the main advantages of topology-based access control is its flexibility in terms of policy specification, as authorized OSN users can be simply specified by stating conditions on relationships, their depth, and trust level. This flexibility,

should be, however, may potentially lead users to lose control of their data. Since access control policies/privacy preferences specify authorized users at an intensional level, i.e., as constraints on relationships in the OSN, the user specifying the rules might not be able to precisely identify who is authorized to access his/her resources or personal data. Even in small social networks, one can hardly understand which users are actually authorized even by simple access control policies such as "friends of friends of my friends", due to the many relationships that users can establish. This possible loss of control generates serious potential risks of unauthorized information flow. Users do not directly know the set of users authorized by their access control policies/privacy preferences, so they actually cannot be aware of their potentially malicious behaviors in releasing accessed data to unauthorized users. Therefore, an interesting issue is to develop techniques and methods to quantify the *potential risks* that may result from the access control policies/privacy preferences specified by OSN users, or from establishing a new relation, so that users are fully aware of the possible effects of their decisions wrt the disclosure of confidential information. The risk measure can also be dependent on the shared resources. For instance, the risk of disclosing different portions of a user OSN profile to another user may be different, dependent on the sensitivity of the information it contains. To the best of our knowledge this is an open issue and no work addressing it has been so far proposed. To cope with this problem, many different strategies can be devised, such as for instance probability-based approaches, where the risk can be estimated based on the probability of propagation of the information associated with each direct relationship in the OSN. Alternatively, methods for information flow prediction can be used to estimate how the risk of unwanted information flow dynamically changes in an OSN. A work which is somehow related to the problem of risk estimation is the one in [2], where a privacy-preserving tool is proposed to enable a user to visualize the view that other users have of his/her Facebook profile, on the basis of the specified privacy settings. However, the user should explicitly select one of his/her neighbours *n* in the OSN to see what *n* can see of his/her profile. Due to the huge number of users in an OSN, it may be almost impossible by using this tool to understand the effect of a policy in terms of unauthorized information disclosure.

15.4 Conclusions

OSNs are today one of the most relevant phenomena related to Web 2.0 and therefore the protection of the information shared through them is a primary need. In this chapter, we have first briefly reviewed the state of the art in the field of OSN access control and then we have discussed some of the most challenging research directions for what concern privacy and access control. One of the fundamental question considered in this chapter is related to the modelling and use of trust. Indeed, one of the key parameter on which information sharing in an OSN should be based is the trust the resource owner has in the recipient users. Open questions

so far not deeply explored are therefore, which is the semantics of trust when trust is used to enforce a controlled information sharing? And how trust can be measured and monitored in a dynamic and complex environment like OSNs?

Acknowledgments The work reported in this paper is partially funded by the Italian MIUR under the ANONIMO project (PRIN-2007F9437X).

References

1. Ali, B., Villegas, W., Maheswaran, M.: A trust based approach for protecting user data in social networks. In: Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative Research (CASCON'07), pp. 288–293 (2007)
2. Anwar, M.M., Fong, P.W.L., Yang, X.-D., Hamilton, H.J.: Visualizing privacy implications of access control policies in social network systems. In: Proceedings of the 4th International Workshop, DPM 2009 and Second International Workshop, SETOP 2009, pp. 106–120 (2009)
3. Bonchi, F., Ferrari, E. (eds.): Privacy-aware Knowledge Discovery: Novel Applications and New Techniques. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series (2010)
4. Bonneau, J., Preibusch, S.: The privacy jungle: on the market for data protection in social networks. In: Proceedings of the Eighth Workshop on the Economics of Information Security (2009)
5. Braghin, S., Ferrari, E., Trombetta, A.: Combining access control and trust negotiations in an on-line social network. In: Proceedings of the Sixth International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010), Chicago, IL, USA, October 2010
6. Carminati, B., Ferrari, E.: Privacy-aware collaborative access control in web-based social networks. In: Proceedings of the 22nd IFIP WG 11.3 Working Conference on Data and Applications Security (2008)
7. Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: Semantic web-based social network access control. Computer & Security (in press)
8. Carminati, B., Ferrari, E., Perego, A.: A decentralized security framework for web-based social networks. Int. J. Inf. Secur. Priv. **2**(4):22–53 (2008)
9. Carminati, B., Ferrari, E., Perego, A.: Enforcing access control in web-based social networks. ACM Trans. Inf. Syst. Secur. **13**(1):1–38 (2009)
10. Castelfranchi, C., Falcone, R.: Trust Theory: A Socio-Cognitive and Computational Model (Wiley Series in Agent Technology), Wiley (2010)
11. Chen, L.: Facebook's feeds cause privacy concerns. The Amherst Student. http://www.halogen.note.amherst.edu/astudent/2006-2007/issue02/news/01.htm. Accessed Oct 2006
12. Domingo-Ferrer, J., Viejo, A., Sebé F., González-Nicolás, I.: Privacy homomorphisms for social networks with private relationships. Elsevier BV, Netherland (2008)
13. Elahi, N., Chowdhury, M.M.R., Noll, J.: Semantic access control in web based communities. In: Proceedings of the Third International Multi-Conference on Computing in the Global Information Technology (ICCGI 2008), pp. 131–136. Washington, DC, USA, IEEE Computer Society (2008)
14. Fong, P.W.L., Anwar, M.M., Zhao, Z.: A privacy preservation model for facebook-style social network systems. In: Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS 2009). Saint-Malo, France, 21–23 Sept 2009
15. Ganjali, Y., Tootoonchian, A., Saroiu, S., Wolman, A.: Lockr: better privacy for social networks. In: Proceedings of the 5th ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT). Rome, Italy (2009)

16. Golbeck, J.: Computing and applying trust in web-based social networks. PhD thesis, College Park, MD, USA (2005)
17. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Discovering leaders from community actions. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08). Napa Valley, California, USA (2008)
18. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: a semantic web rule language combining OWL and RuleML. W3C Member Submission, World Wide Web Consortium. http://www.w3.org/Submission/SWRL. Accessed May 2004
19. Liu, K., Das, K., Grandison, T., Kargupta, H.: Privacy-preserving data analysis on graphs and social networks. In: Kargupta, H., Han, J., Yu, P., Motwani, R., Kumar, V. (eds.) Next Generation Data Mining, pp. 419–437. CRC Press (2008)
20. Mezzour, G., Perrig, A., Gligor, V., Papadimitratos, P.: Privacy-preserving relationship path discovery in social networks. In: Proceedings of the Eighth International Conference on Cryptology and Network Security (CANS 2009), December (2009)
21. Mika, P.: Social Networks and the Semantic Web (Semantic Web and Beyond), 1st edn. Springer, New York, NY (2007)
22. Nin, J., Carminati, B., Ferrari, E., Torra, V.: Computing reputation for collaborative private networks. In: Proceedings of the 33rd IEEE International Computer Software and Applications Conference (COMPSAC) (2009)
23. Squicciarini, A.C., Paci, F., Bertino, E., Trombetta, A., Braghin, S.: Group-based negotiations in P2P systems. IEEE Trans. Parallel Distrib. Syst. **99** (2010) (preprints)
24. Villegas, W., Ali, B., Maheswaran, M.: An access control scheme for protecting personal data. In: Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust, pp. 24–35. Washington, DC, USA, IEEE Computer Society (2008)
25. Winsborough, W.H., Li, N.: Towards practical automated trust negotiation. In: Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002), June 2002
26. Yeung, C., Liccardi, I., Lu, K., Seneviratne, O., Berners-Lee, T.: Decentralization: the future of online social networking. In: W3C Workshop on the Future of Social Networking (2009)

Chapter 16
Dynamic Resiliency to Changes

Fabio Massacci, Federica Paci and Olga Gadyatskaya

Abstract Nowadays IT systems are undergoing an irreversible evolution: we face a socio-technical system where humans are not just "users" but *decision makers* whose decisions determine the behavior of the system as a whole. These decisions will not necessarily be system supported, nor planned in advance and sometimes not even informed, but they will nonetheless be taken. Thus, the inclusion of humans as decision makers requires the provision of technical means so that organizations can balance the need of getting the work done in presence of changes to the original course of action without incurring each and every time the risk of unforeseen toxic over-entitlements. In this paper, we consider a particular case of change to the course of action that is when the users availability or the assignment of privileges to users changes. When such changes occur the satisfaction of organizational business goals can no longer be guaranteed: in fact business goals are achieved through the execution of organizational business process activities and if there are no user that can execute the activities or users with the right privileges and responsibilities, business goals cannot be satisfied. We refer to this problem as *dynamic resiliency* to changes in the assignment of users to roles.

Keywords User roles · Social networks · Permissions

F. Massacci · F. Paci (✉) · O. Gadyatskaya
DISI, University of Trento, Trento, Italy
e-mail: paci@disi.unitn.it

F. Massacci
e-mail: massacci@disi.unitn.it

O. Gadyatskaya
e-mail: gadyatskaya@disi.unitn.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_16,

16.1 Introduction

Nowadays IT systems are undergoing an irreversible evolution: we no longer have a purely technical system whose design and behavior can be predicted with purely technical methods. We face a socio-technical system (STS for short) where humans are not just "users". They are *decision makers* whose decisions determine the behavior of the system as a whole. These decisions will not necessarily be system supported, nor planned in advance and sometimes not even informed, but they will nonetheless be taken.

The parallel technical development of service-oriented architectures, remote privilege management infrastructures, remote maintenance and service outsourcing has distributed and multiplied the points in which humans interact with the STS and decide its behavior. The rationale and the possibility itself of the decisions by the staff in the radiology ward are likely unknown to the people in the IT department.

This is a major challenge for security research as not all actions are authorized and not all infringements are done for the achievement of the organizational objectives. The known mechanisms of escalating requests for authorization to more powerful entities in the system is not usable even for personal users (the Yes/No pop-up windows), and do not scale to the full complexity of STS. Sinclar et al. (FinanceCom'2007) in a field study on information security risks in finance noted that most corporations no longer have direct managers but figures such as group manager, engagement manager, etc.

For example, in the spring of 2009 a professor of computing and a father was complaining at a radiology ward. A CD with X-rays of his son's chest had garbled images. Unfortunately, the CD burning process has been outsourced and, in compliance with e-health security policies, technicians could not see the images on the system. Only doctors could. Thus the staff had a decision to make: sidestep the father (send him away with empty hands to the pneumology ward) or sidestep the system (give the technician the doctor's password and thus the ability to access all images and not just this one)? Who should have granted the authorization? The head of radiology? the head of pneumology? The head of IT department? The head of personnel?

The key word here is "change". The context for which the STS was designed has changed and we need to cope with it. We cannot collect all possible trade-off between security and functionality, nor spell out all possible ways to resolve conflicts. The S-side of STS can manage changes: we know that we might need to do other things than originally stipulated (i.e. infringements) because things change but goals stay the same.

The scientific challenge is that security research and technology on the T-side is significantly less adaptive. While significant research exists on enforcement formal models and techniques (e.g., Polymer, PSLAng, and Security-by-Contract, Kirin, UCON, OrBAC, etc.), and significant implementation and technology transfer effort (e.g., the EU IST-projects POSITIF, PRIME, S3MS, MASTER, MOBIUS, PRIME[LIFE] SECURECHANGE) *access control models and mechanisms of*

enforcement are permeated by the idea that infringements are violations and as such should not be permitted. Whenever infringements are permitted, this permission is totally assigned to the human components without supporting her judgments.

We must provide technical means so that organizations can balance the need of getting the work done (send a patient to the pneumology ward with the correct diagnosis) in presence of changes to the original course of action (the garbled X-rays) without incurring each and every time the risk of unforeseen toxic over-entitlements (the give away of the doctor's password). The practical problem is evident, and we shall see that its solution is really a problem with a ground breaking impact on science and society.

In this paper, we consider a particular case of change to the course of action that is when the users availability or the assignment of privileges to users changes. When such changes occur the satisfaction of organizational business goals can no longer be guaranteed: in fact business goals are achieved through the execution of organizational business process activities and if there are no user that can execute such activities, goals cannot be achieved. We refer to this problem as *dynamic resiliency*. When users' assignment to roles changes during the execution of a business process, we want to find the "right" user to perform the activities whose execution is still pending. This means finding a user who is assigned to a role that is entitled to execute the pending activity and is responsible for the achievement of the goal fulfilled by the execution of the activity.

To check whether a business process instance is dynamically resilient, we model an organization as an hypergraph where the vertexes represent the organizational business goals, the activities that implement such goals, the organizational roles and the users assigned to these roles. Then, for each activity we traverse the graph to find those paths that have the activity as initial vertex and go back to the activity by traversing the node representing a goal fulfilled by the execution of the activity, and the nodes representing a user, and a role which is both responsible for fulfilling the goal and it is authorized to execute the activity.

The reminder of the paper is organized as follows. Section 16.2 discusses the state of the art. Section 16.3 introduces the running example. Section 16.4 gives an overview of our approach to verify that a business process instance is dynamically resilient. Section 16.5 concludes the paper and outlines future research directions.

16.2 State of the Art

The problem of how to include humans as additional participants of a SOA-based business process and how to verify the authorizations users have on the execution of business process activities is gaining attention.

BPEL4People [1] is a recent proposal to handle person-to-person WS-BPEL business process. With respect to our proposal, in BPEL4People users that have to perform the activities of a WS-BPEL business process are directly specified in the

process by user identifier(s) or by groups of people's names. No assumption is made on how the assignment is done or on how it is possible to enforce constraints like separation of duty.

Koshutanski et al. [2] have proposed an RBAC model for business processes based on Web services. The model of Koshutanski et al. supports also the specification of authorizations constraints on the set of users and roles. They consider the problem of how to enforce authorizations on the execution of business process's activities. An authorization decision is taken by orchestrating the authorization processes of each Web service, the activities of which are orchestrated in the business process.

Xiangpeng et al. [5] propose an RBAC access control model for WS-BPEL business process. Roles correspond to `partnerRole` elements in the WS-BPEL specification and are organized in a hierarchy. Permissions correspond to the execution of the basic activities in the process specification. In addition, separation of duty constraints can be specified.

The proposals of Wang et al. [4] and Paci et al. [3] are the one more closely related to our work. Wang et al. investigate the resiliency problem in workflow systems, and propose three different notions of resiliency and investigate the computational complexity of checking resiliency. In static resiliency, a number of users are absent before the execution of a workflow instance, while remaining users will not be absent during the execution; in decremental resiliency, users may be absent before or during the execution of a workflow instance, and absent users will not become available again; in dynamic resiliency, users may be absent before or during the execution of a workflow instance and absent users may become available again.

Paci et al. have investigated the static resiliency problem in the context of RBAC–WS-BPEL, an authorization model for WS-BPEL that supports the specification of authorizations for the execution of WS-BPEL process activities by roles and users, authorization constraints, such as separation and binding of duty, and resiliency constraints that specify the minimum number of users that have to be available for the execution of an activity. They propose an algorithm to determine if a WS-BPEL process is statically resilient to user unavailability.

With respect to Wang et al. and Paci et al. proposals, in this paper we have proposed an approach to investigate that a business process instance is dynamically resilient not only to users unavailability but also to users changing their role.

16.3 The Drug Reimbursement Process

In this paper we use as illustrative example a business process adapted from the MASTER project http://www.master-fp7.eu/ for drug reimbursement called File F process shown in Fig. 16.1. It consists of four macro-phases executed by different actors. The first phase is the drug prescription done by a doctor to a patient to care patient's disease, while the second phase is the drug dispensation to the patients. During the third phase a number of reports are generated and audited. The last

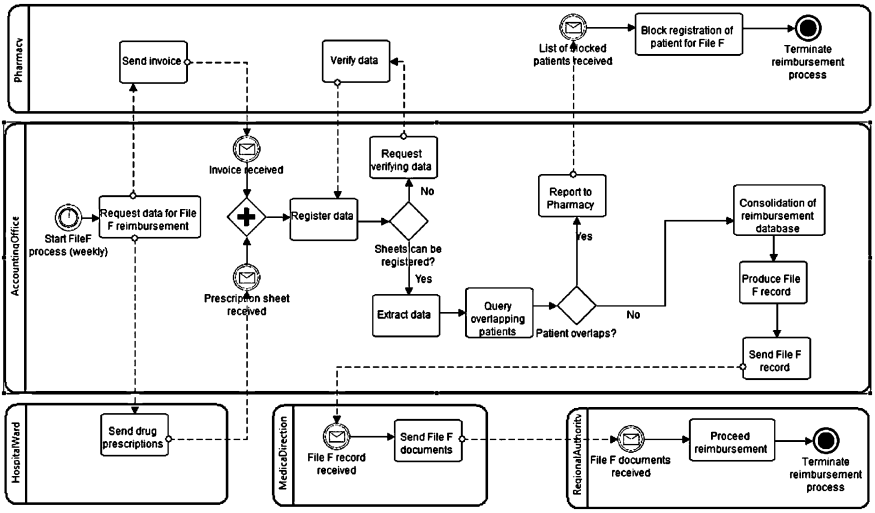


Fig. 16.1 File F business process

phase is about cost reimbursement by local health authorities. Here we focus on the third phase which involves multiple actors, under different trust domains: the Hospital Ward, the Pharmacy, the Accounting Office, the Medical Direction, and the Regional Authority. The Hospital Ward sends drug requests and notifies the pharmacy about the drug availability in the ward stock; it is also responsible to send the dispensation documents to the Accounting Office. The Pharmacy is responsible for the update of all drug data (price, codes, etc.). The Accounting Office performs the File F data extraction and generates File F records that are later sent to the Regional Authority. The Medical Direction has to send the File F records to the Regional Health Care Authority.

In this process, people can change role. For example doctors can be promoted to the medical direction or can leave the hospital to work for their own practice or another hospital. Administrative staff can be re-assigned to the planning or accounting. No matter what happens, at the end of the month the hospital must be able to complete the process for drug reimbursement. At the same time the presence of significant privacy, security and medical regulations the hospital as whole cannot just assign the first available user to a task in order to conclude a subprocess. The user must have enough the privileges to do so.

16.4 An Approach to Dynamic Resiliency

The assignment of users to roles dynamically changes over time in an organization. Users can permanently be assigned to a new role as effect of a promotion or temporary assigned because of an emergency situation. Users can also become

unavailable because they get sick, they are overwhelmed by work or they go on holiday.

When the assignment changes during the execution of a business process instance, there might not be assignments of users for the activities not executed yet. As a consequence, the business process instance does not terminate and some of the organizational business goals are not satisfied causing a damage to the organization.

Informally a business process instance is *dynamically resilient* to changes in the assignment of users to roles, if for each activity A_k that has not been executed yet, there is at least a *potential owner*. A potential owner is a user who is assigned to a role that (a) is authorized to execute the pending activity without violating authorization constraints, and (b) is responsible for the achievement of the goal fulfilled by the execution of the activity.

Our approach consists in computing for each activity A_k in a business process instance BP_i, all the possible potential owners that can execute A_k.

To find all the possible potential owners, we, model an organization as an acyclic hypergraph. The vertexes represent the organizational business goals, the activities that implement such goals, the organizational roles, and the users assigned to these roles. The hyperarcs represent different type of relations between the vertexes:

- `Decompose`, the *Decomposition* relation which associates with a goal G its subgoals $G_1, G_2, \ldots G_n$.
- `Impl`, *Implementation* relation. It associates goals with the business process activities that implement them.
- `PA`, the *Roles to Activities Assignment* relation, and its symmetric relation $\texttt{PA}^*$. `PA` relation associates with roles the business process activities that roles are authorized to execute.
- `UA`, the *Users to Roles Assignment* relation. This relation associates with roles the users who take on the roles.
- `Resp`, the *Goals to Roles Assignment* relation. This relations assigns to roles the goals they are responsible to achieve.

Formally, finding all the possible potential owners for an activity A_k means traverse the organizational model and find the hyperpaths $A_k \rightarrow G \rightarrow R' \rightarrow U' \rightarrow R' \rightarrow A_k$ such that A_k is the initial and final node and traverses the nodes G, R', U' and R' where G is a goal fulfilled by the execution of A_k, R' is a role who has to achieve goal G and is authorized to execute A_k, and U' is a user assigned to the role R'. For each path we store the tuple $\langle U', R', A_k \rangle$ that corresponds to a potential owner for the activity A_k.

During the execution of business process instance, the set of potential owners for each activity is dynamically updated every time an activity is executed or when the assignment of users to roles changes. If for some of the activities not performed

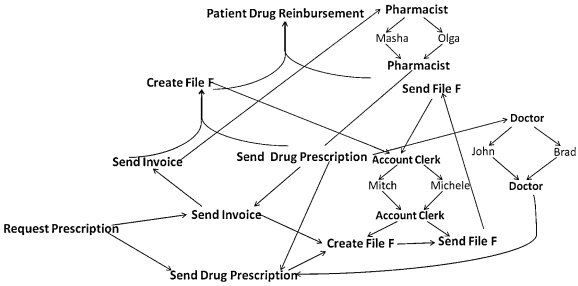


Fig. 16.2 Hospital organizational model

yet, there is no potential owner such that any authorization constraint is violated, the business process instance is not resilient.

Let's now illustrate our approach with an example.

Example 1 Figure 16.2 illustrates the organizational model of an hospital in the North of Italy where the File F business process is deployed. The top goal of the hospital is "Patient Drug Reimbursement" that is decomposed in two subgoals "Send File F" and "Create File F" which on its turn is decomposed in the goals "Send Invoice", and "Send Drug Prescription". The goals "Send File F", "Create File F", "Send Invoice", and "Send Drug Prescription" are satisfied when the corresponding activities are executed. The set of roles inside the hospital consists of the roles "Pharmacist", "Doctor", and "Account Clerk": "Pharmacist" is responsible for the fulfillment of the goal "Send Invoice" and it is authorized to execute the activity "Send Invoice"; "Doctor" has to satisfy the goal "Send Drug Prescription" and it is granted the execution of activity "Send Drug Prescription"; "Account Clerk" has to fulfill the goals "Send File F" and "Create File F" and is authorized to perform the activities "Send File F", and "Create File F". Assume that there is a separation of duty constraint $SoD_u(CreateFileF, SendFileF)$ between the activities "Create File F" and "Send File F". Moreover, assume that the execution of an instance of the File F business process has started and that the activities "Send Invoice", "Send Drug Prescription", and "Create File F" have been executed by the users Masha, John and Michele respectively. Thus, the set of potential owners for the activities *Send Drug Prescription*, *Send Invoice*, *Create File F*, and *Send File F* are updated as follows: {John, Pharmacist}, {Masha, Doctor}, {Michele, Account Clerk}, {Mitch, Account Clerk}.

During the execution of the activity "Create File F", the user Mitch becomes unavailable. Thus, since Mitch is the only user who can perform the activity "Send File F" because of the separation of duty constraint between activities "Create File F" and "Send File F", but he is not available the instance of the File F business process is not resilient to the changes in UA relation

16.5 Conclusions

In this paper, we have investigated the problem of *dynamic resiliency* to changes in the assignment of users to roles. We have proposed an approach to verify that when there are changes in the assignment of users to roles, the execution of a business process instance can still terminate. The approach is based on finding a potential owner for those activities the execution of which is still pending. The potential owner is a user who is assigned to a role that is entitled to execute the pending activity and is responsible for the achievement of the goal fulfilled by the execution of the activity.

To check whether a business process instance is dynamically resilient, we have modeled an organization as an hypergraph where the vertexes represent the organizational business goals, the activities that implement such goals, the organizational roles, and the users assigned to these roles. Then, for each activity we have traversed the graph to find those paths that correspond the activity's potential owners.

We are planning to extend this work in several directions. First, we would like to consider the problem of *escalation*. Escalation takes place if an activity does not meet its modeled time constraints. If this occurs, a notification is sent to the users responsible for the fulfillment of the goal the achievement of which depends on the execution of the activity. We also want to represent the changes in the users to roles assignment relation as graph patterns and apply graph transformation techniques to automatically update the organizational model on the basis of which we evaluate whether a business process instance is dynamically resilient.

References

1. Agrawal, A., et al.: WS-BPEL Extension for People (BPEL4People), Version 1.0. (2007) http://www.adobe.com/devnet/livecycle/pdfs/bpel4people_spec.pdf
2. Kostutanski, H., Massacci, F.: An access control framework for business processes for web services, In ACM Workshop on XML Security, pp. 15–24, George W. Johnson Center at George Mason University, Fairfax, Va, USA, October (2003)
3. Paci, F., Ferrini, R., Sun, Y., Bertino, E.: Authorization and user failure resiliency for ws-bpel business processes. In: Sixth International Conference on Service Oriented Computing (ICSOC), (2008)
4. Wang, Q., Li, N.: Satisfiability and resiliency in workflow systems. In: Proceedings of ESORICS, pp. 90–105 (2007)
5. Xiangpeng, Z., Cerone, A., Krishnan, P.: Erifying bpel workflows under authorisation constraints. In: Proceedings of Fourth International Conference on Business Process Management (BPM 2006), Vienna, Austria, September (2006)

Chapter 17
Certifying Security and Privacy Properties in the Internet of Services

Marco Anisetti, Claudio A. Ardagna and Ernesto Damiani

Abstract Certification is a well-established approach for the provision of assertions on security and privacy properties of entities (products, systems, services). People using (or other entities interacting with) certified entities can rely on the asserted properties, provided that the process of certification is known to produce sufficient evidence for the validity of the property for the certified entity. Today, business processes are increasingly implemented via run-time selection and composition of remote components provided by service suppliers. On the future Internet of Services, service purchasers will like (i) to have certified evidence that the remote services possess some desired non-functional properties, including service security, reliability, and quality, (ii) to be able to infer process-level properties across certified services' composition. In this chapter, we provide a first analysis of the challenges to be faced toward security certification in the Internet of services, outlining possible solutions and future research directions.

Keywords Service composition · Security · Remote services

M. Anisetti, C. A. Ardagna (✉) · E. Damiani
Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano,
Via Bramante 65, 26013 Crema, Italy
e-mail: claudio.ardagna@unimi.it

M. Anisetti
e-mail: marco.anisetti@unimi.it

E. Damiani
e-mail: ernesto.damiani@unimi.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_17,

17.1 Introduction

In this chapter, we shall explore the role of certification in tackling the challenges of the Future Internet environment from a security and trust perspective. Besides infrastructure components and protocols, the Future Internet will be composed of active content (e.g., scripted resources) and integration means (e.g., mashups and service composition environments). Current research trends [8, 27] show that functionality in the Future Internet will be provided by means of services. It is widely acknowledged that such a complex environment will have much stronger requirements on providing advanced techniques for assurance[1] and trustworthiness.

In this chapter, the notion of service will be used in a broad sense, to refer both to an interoperability paradigm, as for instance Web Services do provide, and to a business meaning, as service offerings that can be leveraged towards a new dimension of Digital Ecosystem (i.e., The Internet of Services). It is assumed that the typical functions of Service-Oriented Architectures, like orchestration, choreography and instrumentation [25], will be provided on top of the Future Internet.

Today, there is a very broad consensus that transparency of service suppliers' assurance practices and accountability for lack thereof are essential to ensure collective trust in the Future Internet's services. Trust establishment cannot be left to autonomous bilateral negotiation mainly due to a problem of scale: while today trust is to a large extent a relation between a small number of parties, the Future Internet requires schemes supporting a high number of service suppliers and users. This change of scale is due to the fact that trusted interactions will occur at multiple levels: at the level of protocol channels, of individual services, and of communities of individual users. On the Future Internet the *dynamics* of trust will also be important. For instance, if changes in a service composition occur at runtime, how will they affect the user's trust in the composed service? What if these changes are not immediately visible to the composite service's user? We argue that there is a need for developing (and agreeing upon) new trust models for ensuring confidence in participation to the Future Internet, by setting up a (trusted) process to produce and check *certificates* showing that offered services have a given level of assurance. Both testing and formal verification techniques are relevant to these models, that also need to identify the responsibilities of all participating parties, formalizing the services' assurance processes and the related accountability chains. Related challenges involve the architecture (e.g., a selection interface for services taking into account their assurance) and security challenges on how to prevent attackers (e.g., insiders) from tampering with assurance-related metadata.

In the remainder of this chapter, we first present a trust model for a certification-aware Internet of services. We then discuss the problem of certifying security properties of service-based systems [12], presenting some preliminary ideas on

[1] Note that software assurance involves different activities carried out throughout the software development process. The outcome of these activities is made known to the user via written documentation that, in some cases, may be certified by an accredited, trusted third party.

certification of services and analyzing the problem of certifying dynamically composed services, starting from the composition of the certificates of their basic components.

17.2 A Trust Model for Certification

Currently available certification schemes aim to provide trustworthy evidence that a particular software system has some features, conforms to specified requirements, and behaves as expected [10]. In this chapter, we start by providing the simple trust model underlying current certification schemes [9]. In the simplest case, software purchase involves two parties, a supplier s and a purchaser p (see Fig. 17.1a). Suppliers make claims, i.e., publish assertions about their software systems' functionalities, as well as on their non-functional properties. Here we distinguish two types of such assertions: assertions on functionalities of a systems S, in the form S *has functionality* F (e.g., "*the system supports CHAP challenge-based password exchange*"), and assertions linking functionalities and some desired abstract properties, in the form *Functionalities* $F_1, \ldots, F_n$ *imply property* P (e.g., "*CHAP challenge-based password exchange implies password security-in-transit*"). Let us focus on assertions of the first type, as these are the ones that are usually certified. In the following, we will denote with A_e assertions made by an entity e and with E_e evidences produced by an entity e. In our simple trust model, the purchaser p's trust in an assertion A_s made by the supplier s is denoted $T(p, A_s)$, where T takes discrete values on an ordinal scale, say form 1 to 7.[2] The value of T depends on many factors, including the market standing of s and the evidence that s can provide supporting her assertion, e.g., via shared testing, beta releases, and the like. The purchaser p can also trust the way in which the evidence supporting the assertion has been generated. This is denoted as $T(p, E_s)$. Certification modifies p's trust in assertions of the form "S *has functionality* F" by introducing trusted external entities in charge of collecting, validating and publishing them and the related evidence (see Fig. 17.1b). By doing so, a certification process also introduces new types of assertions that p can trust, describing the collection and validation process. In particular, assertions produced by the security certification process answer the following questions:

- *What security functionalities does the system have?* Certification enables other entities besides the supplier s to make assertions about a software system. Security certifications usually provide assertions A_e of the form "S *has functionality* F", listing the security functionalities F of certified systems. The purchasers' trust in a product will depend on it having all functionalities

[2] Alternatively, T codomain could be the interval $[0, 1]$. For the sake of simplicity, we consider a discrete codomain.

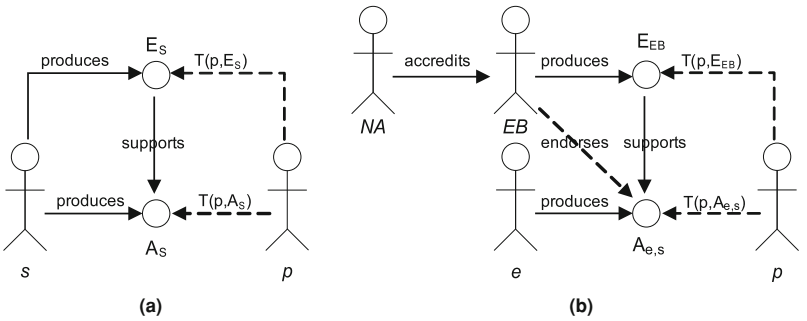


Fig. 17.1 An overview of traditional trust model (**a**) and modified trust model with certification (**b**) of software purchase

required to achieve some abstract security property P, e.g., one belonging to the well-known confidentiality, integrity, or availability classification.

- *Who has validated the system's security functionalities?* The value of $T(p, A_e)$ will depend on the entity e signing the assertion, as well as on the evidence (e.g., a test result or a model based proof) backing it. Certification schemes need to clearly define how each entity e is accredited, e.g., by national authorities that in turn are set up by laws or international treaties, and the signature schemes used to sign assertions. Usually, certifications provide a set of criteria for entities called *security evaluation facilities*, to ensure that these entities are capable and competent of performing security evaluations under a clearly defined quality system.
- *How have the security functionalities been validated?* Assuming that an assertion A_e is signed by a source e that is accredited by an universally trusted authority, the value of $T(p, A_e)$ will only depend on the nature of the available evidence (possibly, signed by e) supporting the assertion. A security functionality F can be tested using different test suites; alternatively, its properties can be proved based on a formal model, which in turn can be part of the design documentation or extracted from the system itself, e.g., by means of code analysis. The focus of the validation can be on security functionality alone, or, the development process may also have been taken into account. Certification schemes clearly define how evidence is to be collected and stored, how the product has been validated.

Fig. 17.1b shows a trust model for certification. The model considers a supplier s, a purchaser p, an entity e producing assertions on s, and a National Authority (NA) that accredits an Evaluation Body (EB) to be trusted to carry out evaluations. An entity e different from s may provide an assertion $A_{e,s}$ on s, on which p has a given level of trust $T(p, A_{e,s})$. Moreover, p specifies a level of trust $T(p, E_{EB})$ on the evidence produced by EB (which implicitly endorses the assertion) taking into account several parameters, as for instance EB's reputation and the mechanisms used to provide the evidence. It is also important to remark that certifications are intended for different target groups. Certificates can be used as a selling argument, compared to a competitor's product with no certificate; in this case, the supplier assumes that $T(p, A_{e,s}) \geq T(p, A_s)$, and that the increase in revenue due to

increased trust will be greater than the cost of certification.[3] This is indeed the case when certificates are recognized internationally and the credibility of the supplier cannot be expected to match the certificate's one. More frequently, and perhaps more importantly, governments set up an accreditation scheme for security products, and mandate purchasers belonging to special categories of customers to buy certified products only. In this case, certificates are used as templates to express compulsory technical security requirements. As a consequence, customers only authorize purchases of certified products, in order to make sure that no purchased system will lack the mandatory security functionalities and above all, to escape liability for the consequences of known security attacks that such functionalities are meant to prevent.

When a business process is enacted by run-time selection and composition of different services, certified evidence of assurance regarding individual services can be used to select the appropriate ones. More ambitiously, a (certifiably correct) inference process can be used to check that the process obtained by composition has some (possibly different) properties. This scenario of compositional, machine-readable certificates to be used at run time is clearly not matched by current software certification schemes, that consider monolithic software and provide human-readable, system-wide certificates to be used at deployment and installation time to support users in the selection of software systems that best fit their requirements.

17.3 Common Criteria

To fix our ideas, we will refer to an existing certification scheme, the Common Criteria (CC) [18]. CC typically includes a thorough examination by experts following pre-defined and publicly accepted criteria, and produce documents and explanations intended for a human user supporting him in the decision on whether to use/buy a system or not. Thus, security certificates are formulated in natural language and address a high abstraction level. Also, the asserted properties are often not even mentioned in the certificate. They are either part of the certification scheme or expressed in a separate document (called Security Target in Common Criteria). CC is a way to define, assess, and validate the security aspects of ICT products. Organizations using security products can define their technical security requirements for a type of product in a Protection Profile (PP). Developers can show compliance of their product to a PP described in the developer's Security Target (ST). CC supports understanding of "what the product does security-wise" and "how sure you are of that". In the CC standard [18], security requirements are expressed in natural language, and their structure is dictated by the corresponding methodology. Furthermore, a certificate refers to a particular version of the product or system. In general, changes in the system structure require re-certification.

[3] The same is valid for the evidence, that is, $T(p, E_{EB}) \geq T(p, E_s)$.

Though CC contains an assurance class on *flaw remediation*, it is rarely used and does not provide methodological support for analyzing the security impact of system changes. Today, the software system in the scope of a certification is nearly always considered to be monolithic. In CC, for instance, the system borders are explicitly defined, and security assumptions on the environment can be expressed. The most recent version of Common Criteria, CC v3.1, allows to deal with composite systems (i.e., derive a system certification from certificates of its components), but requires a perfect match between assumptions and component guarantees. The schemes do not provide support for dynamic changes of components (i.e., at run-time). Even with CC v3.1, changing components would require evaluator/expert interaction and a repetition of (parts of) the evaluation and certification. In addition, the evidence itself is typically not part of the certificate awarded, so that the relying party needs to trust the certificate, the experts, and the certification scheme. This trust is established by the scheme being run by accredited authorities, the accreditation of the experts themselves, and the certificate being officially approved. Here, we claim that the definition of a certification scheme for services is crucial in order to make trusted assurance information available in service-based business process enactment [12].

17.4 A Closer Look to Service Security Certification

Traditionally, research on security properties of services and processes has focused on the protection of communication confidentiality and integrity [17]. Security solutions exist at different protocol levels, including transport layer protection (e.g., HTTPS), application-level authentication (e.g., HTTP digest), message layer protection (e.g., SAML, XML signature and encryption), application layer protection (e.g., XACML, WS-Policy) [4]. However, security properties of interest for individual services are known to have a much wider scope than the above solutions can address [17, 19]. Proceeding bottom-up in the service protocol stack, (i) prevention of malformed or otherwise critical XML answers is important to ensure successful service authentication [7, 26, 29]; (ii) point-to-point non-repudiation is essential for many types of business transactions [24]; (iii) knowing how (and how long) information supplied to a service will be managed is crucial for preventing information leaks [33].

Here we argue that all stating these security properties in a machine-readable form and having them signed by a suitable authority can boost user confidence in using a service. The process owner can define different selection policies used by the process orchestrator during the composition, each one posing restrictions on the security properties a single service should have, and on the way such properties should have been evaluated and certified. The certification of security properties of service-based infrastructures is a fundamental requirement for the definition of processes implemented via runtime selection and composition of individual components. Service composition in fact can be driven by the analysis of certified

properties of services at selection time. Certification of security properties must provide appropriate evidence that the service really supports the properties stated in the certificate and behaves as expected. Two main types of certification processes (and related evidences) are of interest in the considered scenario: (i) *test-based certification* providing test-based proofs that a test carried out on the software has given a certain result, which in turn shows (perhaps with a certain level of uncertainty) that a given property holds for that software; or (ii) *model-based certification* providing model-based proofs that a given service holds some properties and meets formal specifications in terms of security requirements.

A more ambitious goal is then to provide a mechanism that, starting from the properties of the individual services composing a process (and taking into account the orchestration), automatically computes the properties of the process. Indeed, intuition suggests that some properties of individual services could be used to infer properties of processes, at least when the process plan and coordination schema are known. Of course, here we are not arguing that arbitrary properties can be inferred this way from process orchestrations, because of the well-known problems of computational complexity and even decidability that affect run-time verification and validation [23]. Rather, we propose a "properties come first" strategy, where the desired security properties the process owner wants to achieve can sometimes dictate a "safe" service coordination schema that supports their run-time verification or validation, taking into account the available certified properties of individual services and the process timing constraints. When this safe coordination schema cannot be found, supplementary individual properties and/or timing constraints can be suggested to make its computation possible.

17.5 Service-Aware Certification Architecture

A service-aware certification architecture is an infrastructure aimed to support, on one side, security and privacy certification of services and, on the other side, mechanisms that allow process owners to select the services that best fit their requirements. This selection is performed by matching owner preferences with service certificates. The proposed architecture is composed by the following main parties. (i) *Purchaser* (*p*), the process owner that needs to access or integrate a remote service based on the specified preferences. (ii) *Supplier* (*s*), the entity implementing remote services accessed by *p*. (iii) *Evaluation Body* (EB), an independent, trusted tester or accredited lab carrying out evaluations. EB is trusted by both *p* and *s* to run ad-hoc test suites (or generate new ones if needed) and formal checks to evaluate the conformance of the selected services with security preferences expressed in terms of security properties. The evaluation outcome (artifacts) is a pre-requisite for issuing a certificate on the part of the certification authority. (iv) *Certification Authority* (CA), an entity trusted by one or more users to create and assign certificates based on the artifacts provided by EB. Process artifacts enable CA to generate an evidence-based certificate guaranteeing that

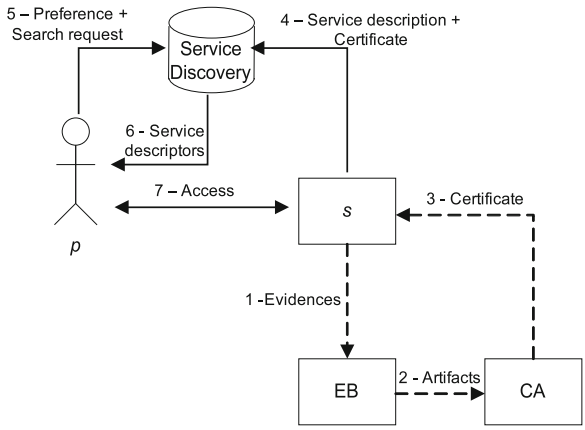


Fig. 17.2 Service-aware certification architecture

some tests and/or formal checks have been performed on a method, a service, or an entire business process in a given context and with a certain result. Also, evidence-based certificates specify which property the tests and/or formal checks are meant to support, and if (and how) the property was used to generate the test-cases and/or formal proofs. (v) *Service Discovery*, a registry of services (e.g., UDDI [32]) enhanced with the support for certificates and preferences.

Figure 17.2 shows a service-aware certification architecture and its two main flows. In the first flow (Steps 1–3), a traditional certification process is presented [18]. The supplier *s* starts the certification process for one of its services by generating and sending evidences, possibly including service code, interfaces, documentation, tests, to EB (Step 1). EB evaluates the service and generates a set of artifacts based on the received evidences (Step 2), which are then used by CA to issue a certificate that proves some properties of the service (Step 3). In the second flow (Steps 4–7), *s* first registers each certified service at *Service Discovery* (Step 4) by sending the description of the service (e.g., a WSDL description) together with its certificate. After the registration, the service is available and can be accessed by *p* as follows: (i) *p* issues a request including a set of preferences on the security properties that must be satisfied and certified by the services (Step 5), (ii) Service Discovery returns a set of descriptors of services that match the stated preferences (Step 6), and (iii) *p* accesses/integrates the best service (Step 7).

The next step in supporting the (semi-) automatic matching of certificates and preferences is the definition of machine-readable certificates. The format of the certificates should be such that they can be simply integrated within current service-based infrastructures and their standard protocols. As a consequence, we put forward the idea of defining an XML-based syntax to define both certificates and preferences. In this way, the certificate can be associated to the WSDL description of the service interfaces, and simply matched with the preferences of *p*.

The certificate should include three fundamental parts: (*i*) an *URI* that links the certificate to the file describing the service (note that also the file may refer

the certificate associated with the service); (ii) *tests* and *models* section that contains artifacts and information about the test-cases and formal checks used in the certification of the service; (iii) the list of certified *properties* with a link to the relevant test-cases and/or formal proofs. Part (ii) of the certificate is the most critical and complex, and needs to be defined to preserve the flexibility of the solution and to manage the heterogeneity of the certification processes. The certificate may also contain a reference to the code of the test-cases and to the formal specifications applied to the service, in order to permit an a-posteriori re-evaluation of the service.

17.6 Certifying Composite Services

Traditionally, research on service composition has focused on two major aspects: (i) ensuring proper interoperability among components and (ii) proving that the composed service will functionally behave as intended. Less attention has been devoted to predicting the non-functional properties of services obtained by composition. Here, we adopt the definition of composition put forward by Milanovic et al. in [22]. Namely,

> *Service composition is the process of binding execution of two or more services, such that functional and non-functional properties can be determined and guaranteed.*

This definition goes well beyond the conventional notion of *orchestration*, i.e., the specification in terms of message exchange of the coordinated execution of multiple services. Indeed, the notion of service composition requires that non-functional properties (e.g., quality of service, security, privacy, dependability) of a composed service can be predicted or estimated based on the ones of its components. Several *model-based* techniques have been proposed to deal with this problem. In principle, one could think of certifying the typing and order of messages exchanged in a service composition by formalizing the composition, e.g., by using a suitable calculus (e.g., π-calculus [21]). However, such a formalization cannot be carried out at run-time, so the composition must be known beforehand. More importantly, a model of each component service as a process is also required; this model must then be provided by the service supplier in order to derive the composition's properties. Also, not all non-functional properties of interest can be expressed this way; for instance the timeliness (or lack thereof) of component invocations would be hard to derive in such a setting. Model-checking looks much more promising for computing properties of service compositions at run-time, having a long history of application to distributed computing [16]. However, properties of compositions cannot always be inferred via run-time model checking because of the well-known problems of computational complexity and even decidability [23]. To avoid complexity issues, one could focus on specific properties (e.g., on service-specific invariants involving local variables and input/output

parameters). The contract-based approach advocated in [22] allows to check simple global properties of a composed service (e.g., expressed as inequalities on the service state variables) by composing component services' invariants (i.e., *contracts*). The approach uses composition operators on contracts to replicate the real service invocation sequence, and obtains the global property as the result.

It is interesting to remark that individual service contracts could be based on test results providing an interesting connection with test-based certifications. Also, the contract-based approach could be used at run-time on a service-based infrastructure, by applying a suitable negotiation scheme to select candidate components whose local contracts support the desired global property [15]. However, while some security properties can indeed be expressed as simple constraints on input or on state variables, this is not the case in general. Deriving at run-time the security properties of an arbitrary composition is much harder than checking a contract.

Therefore, we envision a "properties come first" strategy, where the security properties that the process owner wants to achieve dictate a "safe" service composition schema that supports run-time verification or validation. For example, (usually hard) cardinality-based privacy properties (e.g., the property that *no more than k component services simultaneously hold a given information during each execution*) can be easily imposed for series-only compositions. So, it is conceivable that the need for proving such a property run-time can drive the composition topology, at least when performance is not an issue. To fix our idea, let us consider the following two properties: *invocation parameters are retained by the invoked service for less than 5 ms* and *invocation parameters cannot be inferred from other internal variables or results*. Assume that these two properties are (certifiably) known to hold for all individual services in a simple centralized orchestration.

A simple example of a service holding the latter property is a service computing the arithmetic mean of an array of numbers, where the original input array is immediately obfuscated by the service applying random perturbations to obtain a safe input—later to be used to compute the mean—and then promptly forgotten.[4]

Now, if the process owner knows that the orchestrator will invoke component services in a linear sequence and the orchestration timing (certifiably) shows that invocations are being clocked at 10 ms from each other, the process owner can be sure *at run time* that no information-sharing clique of services can be formed behind his back. In other words, a process-wide security property of *clique avoidance* (i.e., the impossibility of certain information-sharing cliques to arise) is dynamically inferred from individual services' certifications (and the process execution context). When this safe composition schema cannot be found, supplementary individual properties and/or timing constraints can be suggested to make its computation possible.

[4] Other examples of obfuscation can be found in [1–3].

17.7 Related Work

Research on security certification has mainly focused on software components [10]. Security certification of services, instead, is a recent idea [11] and its application is finding several barriers, especially in the intrinsic dynamicity of service composition.

Known concerns in moving toward the Internet of services include ensuring continued compliance to security assurance requirements across re-orchestrations and partial outsourcing of processes. Some preliminary efforts have been made in this direction. In 2008, the US-based Software Engineering Institute (SEI) has published a requirements document on the service certification and accreditation process for the US Army CIO/G-6 [28]. The document describes a process for certifying services in order to assure that they are not malicious to the service-oriented infrastructure that they are deployed in or interacting with. The requirements identified by SEI include shortening existing certification lifecycle: certification of services is expected to be dynamic and services must be certified in a timely way. Applications that dynamically select external services need to be sure that, in doing so, they do not loose security-related properties such as confidentiality, integrity, authentication, and non-repudiation. This level of protection is needed every time information is in storage, processing, or transit on the Internet of services, and no matter whether it is threatened by malice or by accident. The main limitations of SEI are that it focuses on the specific requirements of the military sector alone and does not provide a full architectural support for comparison and negotiation of service certificates. Damiani and Maña [12] have studied the problem of assessing and certifying the correct functioning of SOA and Web services. They introduced a framework together with certificates based on signed test-cases.

In the last few years, other researchers started working on Web service testing. This topic greatly differs from standard testing practices, because the loosely coupled nature of Web services (as compared to traditional component-based software systems) severely limits the way testers can interact with the services during the testing process, making usual stub-based techniques hardly applicable to integration (and security) testing of composite services. In this context, Tsai et al. [31] have proposed a framework aimed at addressing dependability and trustworthiness issues in service-oriented infrastructures that implements group testing to improve test efficiency. A solution based on potency and coverage relationship has been provided for test-case selection and ranking. Kolaczek and Juszczyszyn [20] have defined a method for assessing and evaluating the security level of composed services, in the context of a layered security architecture and multi-agent approach. Research in this direction has also investigated different ways of generating test-cases without the need of accessing the services' source code. Dong et al. [14] have proposed an approach for testing Web services using fault-coverage to check the conformance of the Web services to their WSDL specification, with the goal of automating testing activity. Other works (e.g., [5]) have focused on automatically

generating test-cases starting from the services' WSDL specification, using formal test-case generation techniques. Tsai et al. [30] have proposed an enhanced UDDI server that performs check-in and check-out testing for services. The check-in test is performed when the service is first registered, while the check-out test is done when the service receives a request from a user.

Finally, some works have been done in the context of automatic Web service composition. Berardi et al. [6] have provided *Colombo*, a framework for automatic service composition that considers message exchange, data flow management, and effects on the real world. The proposed solution is based on the concept of goal service that models the expected behaviour of a composite service. Recently, Deng et al. [13] have presented a solution that provides a model for an automatic composition of processes that guarantees correctness constraints, including freeness of deadlock and unspecified receptions, and temporal constraints.

In contrast to the above works, our effort will be aimed to provide a more complete infrastructure where service suppliers may certify the properties of their services via test-based and model-based certification, and machine-readable certificates. Users can then specify their preferences in terms of properties that need to be tested and certified, and automatically retrieve only those services that match the preferences. Finally, our solution will be aimed to provide an approach for certification of composed services starting from the certification of their basic services.

17.8 Conclusions and Future Work

Software systems are increasingly made available as remote services, to be dynamically composed to build complex business processes. In this environment, security and privacy certification of services is a must for improving users trust and confidence in the correctness of the software products/services they adopt. Certification is even more critical when applications to be certified are dynamically built from loosely-coupled, well-separated services. In this chapter, we presented a preliminary discussion of challenges to be studied in the certification of services and composition thereof. Service certification leaves open many research issues that need to be further investigated, such as, the definition of (i) detailed specification of certification practices (environment, runtime context) and outcome, that is, the format of test-based and model-based artifacts to be used in service certification; (ii) a solution to match preferences of the users and certificates of the services for run-time selection; (iii) a mechanism to compare different certificates of the same property; (iv) a solution that allows to certify some properties of a composed service starting from local certificates of its component services.

Acknowledgements This work was partly funded by the European Commission under the project ASSERT4SOA (contract no. FP7-257351). We would like to thank Volkmar Lotz and all partners in the project for their help and fruitful discussions.

References

1. Agrawal, R.: Privacy cognizant information systems. In: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003), Washington, DC, USA, October (2003)
2. Agrawal, R., Evfimievski, A.V., Srikant, R.: Information sharing across private databases. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA, June (2003)
3. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, May (2000)
4. Ardagna, C.A., De Capitani di Vimercati, S.: A comparison of modeling strategies in defining XML-based access control language. Comput. Syst. Sci. Eng. J. **19**(3), 141–149 (2004)
5. Bai, X., Dong, W., Tsai, W.-T., Chen, Y.: WSDL-based automatic test case generation for Web services testing. In: Proceedings of the IEEE International Conference on Service-Oriented System Engineering (SOSE 2005), Beijing, China, October (2005)
6. Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., Mecella, M.: Automatic composition of transition-based semantic Web services with messaging. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005), Trondheim, Norway, August–September (2005)
7. Bhargavan, K., Fournet, C., Gordon, A. D.: Verifying policy-based security for Web services. In: Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS 2004), Washington, DC, USA, October (2004)
8. Cardoso, J., Voigt, K., Winkler, M.: Service engineering for the internet of services. In: Proceedings of the 10th International Conference on Enterprise Information Systems, Barcelona, Spain, June (2008)
9. Chang, E., Hussain, F., Dillon, T.: Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence and Consumer Confidence. Wiley Chichester (2006)
10. Damiani, E., Ardagna, C.A., El Ioini, N.: Open Source Systems Security Certification. Springer, New York (2009)
11. Damiani, E., El Ioini, N., Sillitti, A., Succi, G.: Ws-certificate. In: Proceedings of the IEEE Congress on Services, Part I (SERVICES I 2009), Los Angeles, CA, USA, July (2009)
12. Damiani, E., Maña, A.: Toward ws-certificate. In: Proceedings of the ACM Workshop on Secure Web Services (SWS 2009), Chicago, IL, USA, November (2009)
13. Deng, T., Huai, J., Li, X., Du, Z., Guo, H.: Automated synthesis of composite services with correctness guarantee. In: Proceedinds of the 18th International World Wide Web Conference (WWW 2009), Madrid, Spain, April (2009)
14. Dong, W.-L., Yu, H.: Web service testing method based on fault-coverage. In: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW 2006), Hong Kong, China, October (2006)
15. Dragoni, N., Massacci, F.: Security-by-contract for Web services. In: Proceedings of the 4th ACM Workshop On Secure Web Services (SWS 2007), Fairfax, VA, USA, November (2007)
16. Fu, X., Bultan, T., Su, J.: Formal verification of e-services and workflows. In: Proceedings of the International Workshop on Web Services, E-Business, and the Semantic Web (WES 2002): Foundations, Models, Architecture, Engineering and Applications, Toronto, Canada, May (2002)
17. Galbraith, B., Hankinson, W., Hiotis, A., Janakiraman, M., Prasad, D.V., Trivedi, R., Whitney, D.: Professional Web Services Security. Wrox Press Ltd., Birmingham (2002)
18. Herrmann, D.S.: Using the Common Criteria for IT Security Evaluation. Auerbach Publications, Boca Raton (2002)
19. Jensen, M., Gruschka, N., Herkenhöner, R.: A survey of attacks on Web services. Comput. Sci.—R&D **24**(4), 185–197 (2009)

20. Kolaczek, G., Juszczyszyn, K.: Smart security assessment of composed Web services. Cybern. Syst. **41**(1), 46–61 (2010)
21. Meredith, L.G., Bjorg, S.: Contracts and types. Commun. ACM **46**(10), 41–47 (2003)
22. Milanovic, N., Malek, M.: Verifying correctness of Web services composition. In: Proceedings of the 11th Infofest, Budva, Montenegro, September–October (2004)
23. Necula, G.: Proof-carrying code. In: Proceedings of the ACM Principles of Programming Languages (POPL 1997), Paris, France, January (1997)
24. Papazoglou, M.P.: Web services and business transactions. World Wide Web **6**(1), 49–91 (2003)
25. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. Computer **40**(11), 38–45 (2007)
26. Rahaman, M.A., Schaad, A., Rits, M.: Towards secure SOAP message exchange in a SOA. In: Proceedings of the 3rd ACM Workshop On Secure Web Services (SWS 2006), Alexandria, VA, USA, November (2006)
27. Schroth, C., Janner, T.: Web 2.0 and SOA: Converging concepts enabling the internet of services. IT Professional **9**(3), 36–41 (2007)
28. Securing Web services for army SOA. http://www.sei.cmu.edu/solutions/softwaredev/securing-web-services.cfm
29. Sinha, S.K., Benameur, A.: A formal solution to rewriting attacks on SOAP messages. In: Proceedings of the 5th ACM Workshop On Secure Web Services (SWS 2008), Alexandria, VA, USA, October (2008)
30. Tsai, T., Paul, R., Cao, Z., Yu, L., Saimi, A., Xiao, B.: Verification of Web services using an enhanced UDDI server. In: Proceedings of the 8th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2003), Guadalajara, Mexico, January (2003)
31. Tsai, W.T., Xinyu, Z., Yinong, C., Xiaoying, B.: On testing and evaluating service-oriented software. Computer **41**(8), 40–46 (2008)
32. UDDI OASIS standard. http://uddi.xml.org/
33. Xu, W., Venkatakrishnan, V.N., Sekar, R., Ramakrishnan, I.V.: A framework for building privacy-conscious composite Web services. In: Proceedings of the 2006 IEEE International Conference on Web Services (ICWS 2006), Chicago, IL, USA, September (2006)

Chapter 18
Time-Continuous Authorization of Network Resources Based on Usage Control

Barbara Martini, Paolo Mori, Fabio Martinelli, Aliaksandr Lazouski and Piero Castoldi

Abstract Authorization systems regulate the access to network resources, e.g., bandwidth-guaranteed circuits traversing nodes and links and shared among different media streams, assuring that only admitted data streams use the assigned resources. Traditional access control models were not designed to cope with changes that may occur in the attributes of the user, of the resource or of the environment after the access has been granted. However, in order to prevent misuse and fraud, it is important to extend the control on these attributes after the authorization decision is taken, i.e, during the actual usage of such resources. This control is particularly crucial for network resources because an abuse might cause the degradation of QoS performance for lawful admitted media streams and expose the network to Denial of Service attacks. This paper integrates an authorization system based on the Usage Control model (UCON) in the network service provisioning scenario, to enhance the evaluation of access rights during the actual

This work has been partially supported by the EU FP7 projects: *Context-aware Data-centric Information Sharing* (CONSEQUENCE) and *Open Computing Infrastructures for Elastic Services* (CONTRAIL).

B. Martini
e-mail: barbara.martini@cnit.it

P. Mori · F. Martinelli (✉) · A. Lazouski
Istituto di Informatica e Telematica Consiglio Nazionale delle Ricerche, Pisa, Italy
e-mail: fabio.martinelli@iit.cnr.it

P. Mori
e-mail: paolo.mori@iit.cnr.it

A. Lazouski
e-mail: aliaksandr.lazouski@iit.cnr.it

P. Castoldi
e-mail: castoldi@sssup.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_18,

usage of network resources. The relevant application scenario and architectural design as well as an example of a security policy that implements usage control are described. Finally we outline some open issues and research trends in the applicability of usage control models in networking area.

Keywords Network access models · Access control · QoS

18.1 Introduction

Interactive and bandwidth-greedy applications, e.g., Multimedia On Demand, require the provisioning of network services for transferring media streams with strict QoS requirements in terms of bandwidth, resilience and end-to-end delay. To foster the fulfilment and assurance of these services, resource reservation and access control are required in order to provide secure QoS provisioning that prevents users from injecting traffic load that overloads the allocated resources. Such abuse not only results in the degradation in QoS for other admitted user media flows, but also might lead a system-wide Denial of Service with repercussion on the overall security of the network [1].

In order to regulate the access of users to resources, a number of access control models have been designed, e.g., Discretionary Access Control (DAC) [2], Mandatory Access Control (MAC) [3], Role-based Access Control (RBAC) [4], to verify if a subject, i.e., a user or an application on behalf of him, is allowed to perform an action on an object, e.g., reservation and use of some network resources. Such access control mechanisms are typically implemented in network nodes through Access Control Lists (ACLs) and grant the access to resources based on subject attributes, i.e., identity or credentials presented at the moment of the resource request. Although they represent a significant progress on access control technique, such mechanisms present some shortcomings. The main issue we point out is that they do not prevent the abuse of resources usage from users once the access rights have been granted. In fact, none of them takes into account possible changes of user attributes, resource status or environment conditions and thus they do not verify whether access rights are still valid during the actual usage of the resource [5–7].

This paper proposes an enhancement to the authorization framework for the admission of media streams, based on the Usage Control model (UCON) [7], to enhance the control of network resources, e.g., bandwidth of established circuits traversing nodes and links. Specifically, a policy-based control mechanism has been conceived to enable a time-continuous control of resource usage after the access has been granted. Such mechanism would allow service providers to prevent possible misuse of network resources from undisciplined users while preserving fair resource sharing and profiting from extra-usage by adopting adequate countermeasures (e.g., revocation of resource reservation, over-charging of extra usage).

Hence, the main benefits due to the adoption of the UCON authorization system are: i) dynamic access rights: the rights granted to users are not statically defined, as in the classic access control systems, but they might change over time due to some factors that are evaluated at run-time by the authorization system (e.g., resource availability, user priority, user reputation); ii) time-continuous control: since attributes paired with users and resources can change their values over time, the access right granted to a user at a given time could not hold after some time, even if that access is still in progress. Hence, the security policy is continuously evaluated to revoke accesses when the corresponding rights do not hold any more.

While several research works on time-continuous authorization based on UCON model have been elaborated in the context of Grid Computing [5, 6, 8–10], mobile computing [11] and business transactions [12], to the best of our knowledge, just a resent work has been elaborated for the design of a network resource authorization system. In [13], the authors proposed an extension of the RBAC model that exploits the UCON model to define when the user attribute should be modified in order to grant a privileged role (i.e., quorum role). In this work, the attribute mutability is used to grant system administrators the access to network equipments in order to execute extraordinary maintenance operation in case they do not have enough privileges. Instead, in our work UCON is used to revoke resource usage when the factors that granted the access do not hold any more, thus increasing the overall security of the network.

The paper is organized as follows. Section 18.2 presents a brief overview of the UCON model. Section 18.3 describes the architecture design for network resource authorization including usage control mechanism, while Sect. 18.4 describes an example of usage control policy for the network resource scenario. Section 18.5 discusses advantages and open issues of usage control in networking. Section 18.6 concludes the paper and points to future developments.

18.2 Usage Control

This section gives a brief overview of the Usage Control model (UCON); a detailed description can be found in [7, 14]. UCON is a model that encompasses and extends the existing access control models. Its main novelties are that rights are dynamic, and that subjects' and objects' attributes can be mutable over time, thus requiring continuous enforcement of the security policy during the access time. In the following, we recall the UCON core components: subjects, objects, attributes, authorizations, conditions, obligations.

Subjects and Objects. The subject is the entity that exercises rights, i.e., performs actions on objects. An object, instead, is an entity that is accessed by subjects through access operations.

Attributes. Attributes describe subjects' and objects' features. An attribute is *mutable* when its value is updated as a consequence of accesses performed by subjects on objects, e.g., the reputation. Mutable attributes can be updated before

(*preUpdate*), during (*onUpdate*), or after (*postUpdate*) the execution of the access action. *Immutable* attributes instead, are the classical attributes, that can be updated only through an administrative action, such as the user's identity.

Authorizations. Authorizations are predicates that evaluate subjects and objects attributes and the requested right to take the decision and allow subjects to perform actions on the objects. The evaluation of the authorization predicate can be performed before executing the action (*preAuthorization*), or while the action is in progress (*onAuthorization*).

Conditions. Conditions are environmental or system-oriented decision factors, i.e., dynamic factors that do not depend upon subjects or objects. The evaluation of conditions can be executed before (*preCondition*) or during (*onCondition*) the action.

Obligations. Obligations are used to verify whether the subject has satisfied some mandatory requirements before performing an action (*preObligation*), or whether the subject continuously satisfies these requirements while performing the access (*onObligation*).

Continuous Usage Control. The mutability of subjects' and objects' attributes introduces the necessity to execute the usage decision process continuously in time because, while the access is in progress, the attribute values that previously authorized the access could have been changed in a way such that the access right does not hold any more. In this case, the access is revoked.

18.3 UCON Authorization: Building Blocks and Workflow

This section describes a multimedia service provisioning scenario where a UCON authorization system has been adopted. Specifically, this section details the authorization procedure for the admission of a media stream according to guidelines reported in [15] where the usage control has been included.

The establishment of a media stream implies the management of a session between end-hosts for media flow transfer, and the reservation of the resources across the network so that packet flow related to that session can be treated by network nodes according to QoS requirements. Figure 18.1 depicts the involved entities and the interactions among them to carry out the proposed authorization procedure.

Session Management Server handles user requests and arranges the media transfers to subscribers of their services. The End-Host represents the user device (e.g., VoIP phone, Set Top Box) and comprises a client for requesting network resources to the Edge Routers (e.g., buffer bandwidth on an output interface in the router along the path) by triggering a resource reservation signaling (e.g., RSVP or NSIS) and a client for requesting services (i.e., establishment of a session for a media flow transfer to/from another End-Host or device (e.g., video server, VoIP phone) by triggering a session signaling (e.g., SIP) to the Session Management Server. From the End-Host point of view, the result is the set-up of a network channel for the flowing of the media stream.

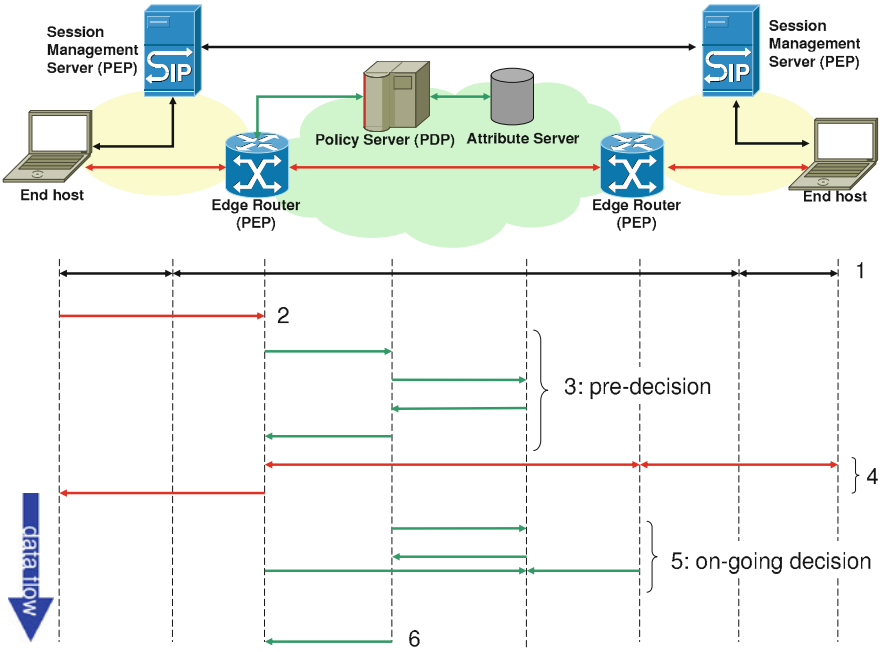


Fig. 18.1 Authorization workflow

The Policy Server is responsible for authorizing the access to services and/or resources according to a usage control policy that exploits the information on resources, users and environment status (e.g., resource availability, user reputation, time) stored in the Attribute Server and periodically updated. Without lack of generality, we assume that the Session Management Server and Edge routers are configured to use the same Policy Server. Then, we suppose that a pre-established trust relationship exists among involved entities in virtue of configuration settings of respective identities and of security keys, i.e., the End-Host is configured to use a Session Management Server associated with the Edge Router to which the End-Host is connected.

According to [16], the system architecture is composed of two main entities: the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). The PDP is the logical entity that decides for authorizing or denying access to services or resources. The PEP is the logical entity that enforces policy decisions taken by the PDP. We can deduce that the network elements that include PDP are the Policy Servers. The network elements including the PEPs are the Edge Routers and Session Management Servers that enforce policies related to the resource usage and to the service delivery, respectively.

When the End-Host issues a request for a set of resources to provide a certain QoS, a policy may be enforced across the network to ensure that resources authorized to a particular flow stay within the bounds of the profile paired to the requesting user, that only admitted flows have access to reserved resources, and

finally that the subsequent usage of such resources do not violate such profile. As an example, a user profile might include the category of subscribed services (e.g., bronze for best-effort Internet access, gold for bandwidth-greedy multimedia service) and the maximum bandwidth allowed for each of them. The policy might include a control on the user reputation depending, for example, on the occurrence of a bandwidth abuse, as detailed in Sect. 18.4.

The message flow for resource authorization procedure with usage control is shown in Fig. 18.1, and is explained in the following. Since we focus on network resource authorization phase, we do not detail the interaction among the Session Management Servers to negotiate the session parameters and to establish the session for media transfer, i.e., session signaling.

1. The End-Host issues a session set-up request to the Session Management Server indicating the media streams requirements. A signaling occurs between Session Management Servers and with the destination End-Host to negotiate media stream parameters to be used in the session (i.e., SESSION SIGNALING). During this interactions the Session Management Server obtains the authorization from a Policy Server through a "token" that then provides to the End-Host in the response to the set-up request.
2. The End-Host issues a request to reserve the resources necessary to provide the required QoS for the media stream (i.e., RESERVE). Included in this request is the token provided by the Session Management Server.
3. The Edge Router intercepts the reservation request and sends an authorization request (i.e., AuthZ REQ) to the Policy Server in order to determine if the resource reservation request should be allowed to proceed. Included in this request is the token from the Policy Server provided by the End-Host. The Policy Server uses this token to correlate the request for resources with the media authorization previously provided to the Session Management Server. The Policy Server, by consulting the Attribute Server, verifies whether the requested resources are in-line with the media streams authorized for the session (i.e., Attr REQ and Attr RSP). Then it sends a decision to the Edge Router. This concludes the traditional access control procedure.
4. If the access request is authorized, the Edge Router triggers an end-to-end resource signaling among routers and destination End-Host to reserve the required resources across the network along the end-to-end path (i.e., RESERVE). Upon receiving the positive response from backwards signaling, the Edge Router sends a response to the End-Host indicating that resource reservation is complete (i.e., RESPONSE). At this point the data stream starts flowing across the network with the required QoS.
5. While the network resources are in use, the Policy Server continuously re-evaluates the security policy exploiting updated value of attributes to check that the right to exploit these resources is still valid. The Attribute Server provides fresh values of users' are resources' attributes to the Policy Server by continuously collecting information from routers and Session Management Servers (i.e., Attr UPDATE).

6. In case a policy violation is detected, the Policy Server revokes the access right (i.e., REVOKE) and, consequently, the data stream stops flowing.

18.4 Usage Policy Example

This section describes an example of a usage control security policy expressed with POLPA language. POLPA is a policy specification language based on process algebra for expressing usage control policies, whose syntax and semantics have been defined in [5, 9].

The policy shown in Table 18.1 regulates the use of network channels created for media flowing. It allows users with GOLD profiles and good reputations to open a number of network channels, provided that the sum of the bandwidths of these channels is below a given threshold. During the usage of the channel, the policy continuously checks the value of the reputation of the user to detect when it goes below a given threshold, and in this case the policy interrupts the usage of the channel. To this aim, the PDP asks the PEP on the Edge router to close the channel.

The policy fragment from line 2 to line 5 allows the creation of a new channel to a given destination with a given bandwidth. The control action *tryaccess* (*cuser*, *net*, *createChannel* (*dest*, *reqBand*, *channel*)) (line 2 of the policy) is received by the PDP from the PEP when it receives the request of the user *cuser* to perform the operation *createChannel*, that creates the new channel on the network. The authorization predicates in line 3 evaluate the attributes of the user to decide whether allowing the access or not. In particular, these predicates check whether the total bandwidth allocated to the user is less than a threshold T, the profile attribute of the user is GOLD, and the reputation attribute of the user is greater than a threshold R. If the previous predicates are satisfied, the control action *permitaccess* (*cuser*, *net*, *createChannel* (*dest*, *reqBand*, *channel*)) in line 4 allows the channel creation, and the user attribute that stores the bandwidth currently

Table 18.1 Example of usage control policy

```
rep(                                                                    1
  tryaccess(cuser, net, createChannel(dest,reqBand,channel)).           2
  {[(cuser.usedBand+reqBand≤T),(cuser.profile=GOLD),(cuser.reputation≥R)].  3
  permitaccess(cuser, net, createChannel(dest,reqBand,channel)).        4
  update(cuser.usedBand=cuser.usedBand+reqBand)}.                       5
  (  (  [ (cuser.reputation<R)].                                        6
       revokeaccess(cuser, net, createChannel(dest,reqBand,channel))    7
    )                                                                   8
    or                                                                  9
    endaccess(cuser, net, createChannel(dest,reqBand,channel))          10
  );                                                                    11
  update(cuser.usedBand=cuser.usedBand-reqBand)                         12
)                                                                       13
```

allocated to the user is updated (line 5). This part of the policy follows the preAuthorization model (as described in [9]) because the authorization predicates are checked before authorizing the access.

The policy fragment from line 6 to 11 implements the continuous control of an authorization predicate because the value of a user attribute, the reputation, could change while the channel is in use. In particular, after the access to the channel is granted (line 4), the policy uses the *or* operator (line 9) to describe two alternative possible behaviours: either the access ends when the user releases the channel (line 10), or it is interrupted while it is in progress because the user reputation is too low (lines 6 and 7). In fact, the predicate in line 6 checks that the reputation of the user is above a given threshold, and it is continuously evaluated when the access is in progress. If this predicate is satisfied, the access is interrupted by the *revokeaccess* control action in line 7 of the policy. Instead, if during the usage of the channel the value of the reputation of the user is greater than R, the resource usage terminates when the user releases the channel, i.e. when the PDP receives the *endaccess* (*cuser*, *net*, *createChannel* (*dest*, *reqBand*, *channel*)) control action from the PEP (line 10). This part of the policy implements the onAuthorization model, where an authorization predicate is continuously checked while the access is in progress. The continuous check of the predicate could be implemented through a subscription mechanism. Each time an access is granted, the PDP could ask the Attribute Server to be notified as soon as the value of the reputation of the user that is performing the access changes. When the PDP is notified, the predicate in line 6 is re-evaluated, and the access is possibly interrupted.

After the termination of the access, the attribute that represents the bandwidth currently allocated to the user is updated (line 12). Finally, the *rep* control action in line 1 of the policy allows the user to open in parallel more channels, provided that the predicates in line 3 are satisfied.

18.5 Usage Control Advantages and Challenges

Over the past years, there have been considerable amount of research on authorization models for a network service admission and usage. However, they mainly provide hard-coded solutions for specific network scenarios (e.g., wireless networks, ADSL networks) [17, 18]. The policy-based authorization framework and the signaling architecture, proposed in this paper, extend existing solutions in many aspects thanks to the included UCON features:

- *Expressive policy model*: The UCON model allows a wide variety of possible policies for network service admission and for usage control to be carried out. It is powerful enough to express priority-based admission control policies, policies based on time-of-day, user identity and credentials, based on some level of "importance", history-based policies, etc [7].
- *Mutable attributes and continuous control*: The UCON model considers that values of attributes paired with users, resources and environment might change

over time due to other accesses performed concurrently by the user the attributes refer to, or even by other users on the network resources. For instance, a single user can initiate several resource provisioning sessions. Changes in attributes might affect the initial conditions when the access was granted. As a result, attributes should be continuously monitored, the policy checks should be performed every time when attributes change, and if new values violate the policy, the access should be revoked and resources should be released.

- *Efficient and secure resource management*: The UCON model releases network resources immediately when a policy abuse is detected by a revocation of granted access rights. This allows to keep the appropriate QoS for eligible users and to prevent Denial-Of-service attacks over the network resources.

Despite the explicit advantages of the UCON model, there are still many issues to be elaborated, such as a more detailed and comprehensive policy model, suitable architectural solutions for heterogeneous networks and, finally, a generic and flexible implementation of usage control security mechanisms.

The policy example given in Sect. 18.4 governs the admission of a media flow with the specified bandwidth (i.e., the channel). Actually, usage control policies should apply to resources at different levels (i.e., the network connectivity level and/or the application service level) as well as on a different scale (i.e., ranging from an entire network to a particular network element like a router, switch, hub, etc.). A Label Switch Path for aggregated traffic across core networks or media streams established on a per-session basis are two examples of different levels of network resources. A buffer in a router could be an example of a fine-grained network resource. Integration of policies on different levels of control should be addressed to preserve consistency as well as architectural elements should be introduced to enforce fine-grained policies. The policy model should specify the unified syntax and types of attributes used to build usage control policies. As initial approach, the attributes for authorization in network resource provisioning discussed in [19] can be adopted.

Further, the advantages of obligations, that can be considered the most meaningful access decision factors in the UCON model, are not discussed sufficiently in domain of network resources. In fact, obligations could strengthen the expressiveness of the policy model and encode into the policy warns like “your request may have to be torn down in 10 min”, “pay to continue an extra usage”, etc.

Regarding architectural issues, the main open problem of the UCON-based policy enforcement is the continuity of control and effective management of mutable attributes. There should exist an active element in the architecture which initiates a querying of up-to-date attribute values and triggers the policy re-evaluation supporting on-going decision process. In [20] several environment-independent models with active PDP, PEP and Attribute Server were considered to facilitate a continuous control. Instead, the network service admission and usage require some specific models due to distributed nature of PEPs what could cause potentially long delays on policy re-evaluation requests that must travel several hops. There is a need of a very accurate model regulating when the policy re-evaluation should take place,

such as re-evaluate every n seconds, or after n packets or n bytes have been transferred, or after other events have occurred in the system. From one side such model should preserve the efficient policy enforcement, and from another side it should minimize the inevitable security overhead in the system. As a matter of fact, a communication protocol between PEP and PDP should be presented which encompasses traditional pre-authorization and continuous control. This requires sufficient extensions to existing signaling protocols (e.g. RSVP, NSIS) to carry authorization information and communications (i.e., exchange on-going access control messages).

Regarding an implementation of security systems, the main issue addresses the lack of flexibility because control mechanisms are typically hard-coded in the application logic and thus suffering the difficulty to be adjusted at runtime. UCON-based systems prevent such issue thanks to a policy-based approach. Nevertheless they introduce an overhead for the processing of security policies that should be minimized. In addition, the scalability issue is of paramount importance because such overhead should not significantly increase with the number of network nodes and of the established sessions. Finally, the survivability of the policy-based authorization systems should be preserved through redundant PDP implementations thus avoiding a single point of failure.

18.6 Conclusions

In this paper, we suggested the adoption of a time-continuous authorization systems based on the UCON model to regulate the usage of network channels for media streams transmission. The contribution of this paper concerns the application of the usage control model to the network resources provisioning scenario, to allow the continuity of authorization decision during the network resources usage. By preventing resource abuse from malicious users, the proposed authorization system further increases the security of the network while assuring QoS to authorized media flows.

As future work, we plan to elaborate on basic implementation of the proposed model and experimentally evaluate the overhead of in terms of policy decision time and in terms of the overall signaling load as a function of the number of network nodes and established sessions.

References

1. Park, F.S., Patnaik, D., Amrutkar, C., Hunter, M.T.: A security evaluation of IMS deployments. 2nd International Conference on Internet Multimedia Services Architecture and Applications (IMSAA 2008), pp. 1–6, 10–12 (2008)
2. Sandhu, R., Samarati, P.: Access control: principle and practice. Commun. Mag. IEEE 32(9), 40–48 (1994)

3. Sandhu, R.: Mandatory controls for database integrity. In: Database Security III: Status and Prospects, pp. 143–150, (1989)
4. Sandhu, R., Coyne, E.J., Feistein, H.L., Youman, C.E.: Role-based access control models. IEEE Comput. 29(2), 38–47 (1996)
5. Martinelli, F., Mori, P.: A model for usage control in grid systems. In: Proceedings of the First International Workshop on Security, Trust and Privacy in Grid Systems (GRID-STP07), IEEE Press (2007)
6. Martinelli, F., Mori, P., Vaccarelli, A.: Towards continuous usage control on grid computational services. In: Proceedings of International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services 2005, IEEE Computer Society, p. 82 (2005)
7. Sandhu, R., Park, J.: The $UCON_{ABC}$ usage control model. ACM Trans. Inf. Syst. Secur. 7(1), 128–174 (2004)
8. Lazouski, A., Colombo, M., Martinelli, F., Mori, P.: On usage control for GRID services. In: Proceedings of the 2009 IEEE International Workshop on HPC and Grid Applications (IWHGA2009) (2009)
9. Martinelli, F., Mori, P.: On usage control for GRID systems. Future Gener. Comput. Syst. Elsevier Sci. 26(7), 1032–1042 (2010)
10. Zhang, X., Nakae, M., Covington, M.J., Sandhu, R.: A usage-based authorization framework for collaborative computing systems. In: Proceedings of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT '06), pp. 180–189 (2006)
11. Castrucci, A., Martinelli, F., Mori, P., Roperti, F.: Enhancing Java ME security support with resource usage monitoring. In: Proceedings of the 10th International Conference on Information and Communications Security (ICICS08). Lecture Notes in Computer Science, vol. 5308, pp. 256–266. Springer, Berlin (2008)
12. Stihler, M., Santin, A.O., Calsavara, A., Marcon, A.L. Jr.: Distributed usage control architecture for business coalitions. In: Proceedings of IEEE International Conference on Communications 2009 (ICC 2009) (2009)
13. Silva, E., Santin, A.O., Jamhour, E., Maziero, C., Toktar, E.: Applying quorum role in network management. In: Proceedings of IFIP/IEEE International Symposium on Integrated Network Management 2009 (IM2009) (2009)
14. Zhang, X., Parisi-Presicce, F., Sandhu, R., Park, J.: Formal model and policy specification of usage control. ACM Trans. Inf. Syst. Secur. 8(4), 351–387 (2005)
15. Hamer, L.N., Gage, B., Shieh, H.: Framework for session set-up with media authorization, IETF RFC 3521, April 2003
16. Yavatkar, R., Pendarakis, D., Guerin, R.: A framework for policy-based admission control. IETF RFC 2753, January 2000
17. Zhi, L., Jing, W., Xiao-su, C., Lian-xing, J.: Research on policy-based access control model. International conference on networks security, wireless communications and trusted computing, vol. 2, pp. 164–167 (2009)
18. Rensing, C., Karsten, M., Stiller, B.: AAA: A survey and a policy-based architecture and framework. IEEE Netw. 16(6), 22–27 (2002)
19. Demchenko, Y.: XACML authorization interoperability profile for network resource provisioning. Phosphorus WP 4 (2008)
20. Lazouski, A., Colombo, M., Martinelli, F., Mori, P.: A proposal on enchancing XACML with continuous usage control features. In: Proceedings of CoreGrid ERCIM Working Group Workshop on Grids, P2P and Service Computing (2009)

Part V
Network Monitoring

Chapter 19
Towards Monitoring Programmability in Future Internet: Challenges and Solutions

Luca Deri, Francesco Fusco and Joseph Gasparakis

Abstract Internet is becoming a global IT infrastructure serving interactive and real-time services ubiquitously accessible by heterogeneous network-enabled devices. In the Internet of Services (IoS) era, monitoring infrastructures must provide to network operators fine-grained service-specific information which can be derived by dissecting application level protocols. To accommodate these new monitoring requirements network probes must be flexible, easy to extend and still be capable of analyzing high-speed network streams. Despite the increased complexity, software and hardware technologies on top of which network probes are implemented have been designed when monitoring requirements were substantially different and almost left unchanged. As a result, implementing modern probes is challenging and time consuming. In this paper we identify desirable features for reducing the work required to develop complex probes, and we present a home-grown comprehensive software framework that significantly simplifies the creation of service-oriented monitoring applications.

Keywords Network monitoring · Programmable probes

L. Deri (✉)
ntop, Pisa, Italy
e-mail: deri@ntop.org

F. Fusco
e-mail: ffu@zurich.ibm.com

J. Gasparakis
e-mail: joseph.gasparakis@intel.com

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_19,

19.1 Introduction

Recent advances in wireless networks and consumer electronics technologies changed the way we are using the internet. The future internet will become a global IT infrastructure providing interactive and real-time services, hence the name internet of Services (IoS), ubiquitously accessible by heterogeneous network-enabled devices.

Understanding service behaviour and measuring the services quality over time is necessary in order to reduce costs while preserving user satisfaction. The quality of service is affected by network performance metrics, such as latency and bandwidth, but also depends on the entire network infrastructure which includes server machines, software and so on. Therefore, network operators and service providers are gradually shifting from a network centric monitoring approach to a service centric monitoring approach that provides a comprehensive and integrated view of the network and allows them to discover the root causes of service quality degradation.

The paradigm shift substantially increased the complexity of monitoring infrastructures. In fact, network probes, which are the key measurement components in today's monitoring architectures, are not only responsible for measuring basic network-level performance metrics (e.g. number of packets), but also for providing detailed service-oriented metrics. Some of these metrics, such as transaction latency, can only be measured by performing flow-level analysis [4] up to the application layer and not by analyzing single packets out of a flow context. As services are often composed of several concurrent communication flows, it is also necessary to correlate them in order to compute service-dependent metrics. For instance, the overall download time of an HTML page has to include the time for retrieving all the external objects (e.g. images) referred from the main page.

The introduction of application layer protocols analysis drastically changed requirements in terms of flexibility, performance, and programmability. Flexibility is required in order to accommodate new requirements (e.g. new protocols) and changed monitoring conditions (e.g. an existing location-fixed service is migrated to a cloud architecture). Performance is necessary for preventing packet drops while coping with the increased packet processing costs due to service-level analysis. Programmability is desirable in order to reduce the work required to extend probes and to adapt them to changing monitoring requirements. As explained in the next section, the rush for performance might have a negative impact on programmability as the use of custom or closed hardware architectures often imposes severe limitations to software applications. As a matter of fact, hardware devices are driving application design and not the other way round, thus jeopardizing flexibility and limiting portability of software applications.

The increasing complexity of monitoring tasks imposed by service-oriented network monitoring did not result in any major evolution of hardware and software monitoring frameworks on top of which network probes are built. This happened because both industries and research communities focused on specific tasks

(e.g. packet capture) rather than on the creation of a comprehensive architecture allowing pluggable modules to be included or replaced in order to satisfy new monitoring requirements. This has been the driving force for the definition of a novel monitoring framework that is modular, programmable by means of software components running on commodity hardware and still capable of exploiting modern hardware technologies.

The rest of the paper is structured as follows. Section 19.2 lists the basic components that monitoring applications require as building blocks, and compares the state of the art of hardware and software frameworks for network monitoring. Section 19.3 describes the monitoring framework we have developed and positions it against similar approaches. Section 19.4 validates the framework against some general monitoring scenarios. Section 19.5 concludes the paper.

19.2 Towards Service-Oriented Network Monitoring

The simplest traffic monitoring application is responsible for capturing traffic and computing packet-based metrics such as the total TCP traffic sent by a specific host. Flow-based monitoring applications, such as NetFlow/IPFIX [5] probes, go beyond this model by adding per-flow metrics which are derived from packet header information. Service-oriented network monitoring applications are capable of providing detailed information about services and not just about network communications. The following paragraphs describe common tasks that makes service-oriented monitoring applications substantially different from the ones listed above.

Payload Inspection. This activity is a prerequisite for properly decoding service primitives. This includes the inspection of tunnels (e.g. GRE and GTP) and encapsulations (e.g. PPTP) as well as the reconstruction of the original encapsulated payload. As of today, packet parsing is usually implemented from scratch in every application, as, beside rare exceptions [11], packet capture libraries such as libpcap [17] do not feature it, or do not release the source code such as NetBee [2] hence limiting their use to selected projects. Commercial libraries such as Hyperscan [14] feature high-speed DPI, whereas QosMOS [20] implements several protocol decoders but tight the application to their closed-source development environments. Wireshark [19] is the richest network protocol analyzer in terms of protocol supported but unfortunately packet decoding and flow analysis code are tight to the application and not available as library, making it unsuitable for integration into applications.

Service-Level Packet Filtering. Most legacy filtering mechanisms such as Berkley Packet Filter (BPF) [18] do not allow the traffic to be filtered by using application-specific fields, whereas other frameworks such as FFPF [3] allow users to define application-specific filters but do not return to applications parsing information nor handle flows. Another limitation of the above technologies is the inability of efficiently adding/removing large number of dynamic filters, which is

necessary for tracking services using dynamic ports such as VoIP and FTP. Contrary to BPF and FFPF, Swift [22] has been designed for offering low latency filter updates, but its scalability in terms of number of configurable filtering rules is limited.

Flow State Persistence. Maintaining protocol state and service-specific performance metrics (e.g. call setup time for VoIP) is necessary for service processing. This increases both processing workload and memory footprint. In addition, service-oriented monitoring applications require scalable, highly efficient and flexible (in terms of data types) storage architectures [13] capable of storing the retrieved service oriented metrics.

Flow Reconstruction. Per-flow packet sequence reordering, defragmentation, and IP datagram reassembly of PDUs spanning across multiple TCP segments must be performed before inspecting service primitives. Performing these tasks substantially increases both packet processing cost and memory footprint, and, therefore it must be enabled only when necessary. In addition, implementing robust and efficient TCP and IP re-assemblers is not trivial [9, 21]. Another important task is to partition the flow into sub-flows whenever several service communications are pipelined over the same connection. For instance in HTTP/1.1 peers can exchange several requests/responses over the same TCP connection.

Packet Capture. Packet loss during capture is not tolerated as it prevents service primitives from being interpreted. Instead, packet and flow-based applications can tolerate limited loss as it leads to inaccurate results while not breaking the overall monitoring system. It is worth noting that in service-oriented monitoring, packet capture is no longer the most resource consuming task, as this is a minor activity when compared to the increased packet processing costs.

Per-Flow Traffic Balancing. Balancing the workload among processing units is necessary in order to leverage modern parallel architectures. When performing service oriented monitoring, packet processing costs depend on the particular traffic to be analyzed and, therefore, balancing packets across units may lead to workload unbalances.

The introduction of service analysis in monitoring infrastructures for high-speed networks raised the demand for flexible monitoring devices capable of speeding up traffic analysis applications. During the years monitoring device vendors focused mostly on performance, neglecting other aspects such as application programmability and portability across different devices designed for traffic analysis acceleration. The lack of common design guidelines across vendors has prevented the creation of a widely accepted and hardware transparent software layer beyond libpcap, which offers primitives limited to packet capture and network device management. Hardware vendors attempted to increase the processing performance in various ways including:

Capture Accelerators. Packet capture accelerators such as DAG cards [10], are special purpose FPGA-based network interfaces that allow the incoming traffic to be captured and copied to the address space of the monitoring application without CPU intervention and without requiring packets to flow through the kernel layers. Often they also provide mechanisms for balancing the traffic among processor

cores and filtering packets, although they are usually limited in features and are not meant to be changed in real-time as they require card reprogramming that may take seconds if not minutes. The main advantage of these hardware devices is the ease of programmability as applications can still run on commodity hardware while significantly improving their packet capture performance. For this reason capture accelerators have been widely accepted by the industry as they represent a simple solution for accelerating traffic monitoring applications, but at the same time they are of limited help in complex monitoring applications. This is because packet capture is no longer the most resource intensive task, and therefore the speed-up achievable with packet accelerators is becoming less significant, but still not marginal.

Strong Multi-Core Systems. Some vendors have embedded strong multi-core systems on network adapters in order to efficiently process packets as close as possible to the network link. Massive multi-core architectures, such as Tilera [1] use a mesh of up to 64 cores embedded directly on the network adapter. The result is that packet capture is no longer a cost as packets are processed directly on the network adapter without the need to move them to the main processor. Another advantage is that the card runs a Linux flavor and that applications can be implemented in standard C, thus significantly easing the development process.

Network Processors. Network processor boards, such as Intel 80579 [15], are special purpose monitoring devices that allow monitoring software to be executed by a processor specifically optimized for packet processing. The emphasis on speed resulted in unconventional hardware architectures providing coprocessors and several packet processing units. Developing applications for network processors is not trivial and requires a deep understanding of low-level architectural details which are usually vendor and model specific. Using external libraries for performing traffic analysis tasks is not always easy either, because applications must be implemented using languages which are similar to C, but not necessarily C compliant.

19.3 A Programmable Network Monitoring Framework

For a few years we have been developing an open-source kernel module for Linux systems named PF_RING [6] that we originally introduced for accelerating packet capture on commodity hardware. Over the years we have realized that it was necessary to go beyond the initial goals and to create a comprehensive framework able to tackle additional issues such as the one listed in the previous section. PF_RING is now a modular monitoring framework that allows developers to focus on implementing monitoring applications without having to deal with low-level details such as packets handling. PF_RING represents an easy to use, yet efficient monitoring framework for developing component based monitoring application. In addition, it provides a hardware transparent filtering mechanism that can be

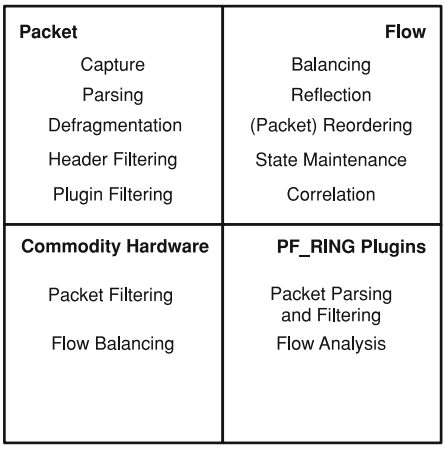


Fig. 19.1 Packet and flow management in PF_RING

eventually accelerated by exploiting features available on modern commodity network adapters.

PF_RING substantially increases packet capture performance. Packets can be captured using standard Linux NIC drivers, but also using capture optimized drivers that allow the kernel to be completely bypassed and modern multi-core processors to be exploited. As of today, we have enhanced 1 and 10 Gbit drivers for popular network adapters by adding support for PF_RING. When running on modern servers and commodity network adapters, PF_RING can capture at wire rate from multiple Gbit links, and over 5 Mpps from 10 Gbit networks [7].

In addition to legacy BPF filters, PF_RING provides more advanced filtering mechanisms that can be used for filtering out unwanted traffic, but also for dispatching packets across analysis components. There are two families of filters: *exact* filters (i.e. all filter fields are specified) and *wild-carded* filters (i.e. at least one filter element has value 'any') where filter fields include MAC address, VLAN, protocol, IP v4/v6 addresses, and application ports. Exact filters are evaluated before wild-carded filters. Contrary to BPF, PF_RING parses the packet, and then checks filtering rules on it. Parsing information is returned as metadata to applications consuming the received packet. Packets are parsed once regardless of the number of consumers and filtering rules. Whenever a filter matches, PF_RING executes the action bound to it. Actions can range from simple packet dropping to more complex operations such as sending packets matching the filter to a network adapter for transmission (a.k.a. packet reflection) (Fig. 19.1).

PF_RING analysis components are plugins implemented as dynamically loadable kernel modules, and identified by a unique numeric identifier that can be associated with one (or more) filtering rule. When a packet matches a rule, the corresponding plugin callback is executed. Developers can define plugin hooks for filtering packets up to layer seven, and forwarding parsing information to the user-space as part of the packet metadata. Therefore, by combining filters and analysis components users can specify L7 filters such as "return only HTTP packets with

method POST", which, contrary to what happens for instance in Wireshark, are executed directly at the kernel layer. Filtering rules can specify a plugin id, thus packets matching such a rule are then passed to the specified plugin for further processing. So far, PF_RING plugins include support for VoIP (Voice Over IP) [12], HTTP, and multimedia streaming.

In PF_RING, flows are used to identify and maintain state for packets matching an exact filter. They can be created automatically by means of plugin actions that are executed whenever a received packet matched. For instance a FTP monitoring application dissecting the control connection by means of a plugin, can add a new exact filtering rule for the tracking data connection as soon as the FTP client initiates a file transfer. For each flow, plugins can keep the state and maintain information about the flow being analyzed. For instance the HTTP plugin maintains information about response code and throughput, whereas the RTP plugin computes jitter and packet loss of voice packets. In a nutshell, the combination of filtering rules and plugins, enables application developers to create powerful monitoring applications by means of simple configuration rules.

PF_RING is implemented as a Linux kernel module that can be compiled without patching the kernel source. A user-space library named libpfring, communicates with the kernel module by means of PF_RING socket and allows applications to transparently interact with the kernel module. Packets are copied by the kernel module into a circular memory buffer that is memory-mapped to user-space. This means that user-space applications can read packets from the buffer without issuing system calls. PF_RING sockets are bound to one physical network interface on which packets are received. As modern network adapters support NIC virtualization and MSI-X (message signaled interrupts), on multi-core systems PF_RING can give applications access to the various virtual RX queues contrary to vanilla Linux, which merges all queues into one. This means that hardware-based mechanisms such as RSS (Receive-Side-Scaling) for balancing network flows among RX queues mapped on processor cores, can be exploited by PF_RING applications to bound to a virtual RX queue in order to receive a portion of the traffic. This solution enables scalability as applications can be partitioned into threads or processes, each bound to a RX queue, that can process a portion of the traffic as highlighted in Fig. 19.2.

In some cases it might be useful to overcome RSS and assign selected flows to a specific RX queue in order to create specialized traffic analysis applications each sitting on a specific queue. In order to achieve this goal, we have recently added into PF_RING support for the latest generation of network adapters such as Intel 82599 controller that allows the driver to force flow balancing to cores by means of a mechanism called flow director (FD) [16]. Binding specific network flows to a non-existing core (e.g. to a core id that is greater than the number of available processor cores) instructs the adapter to drop such flow, hence implementing a wire-speed traffic filter and balancer. PF_RING comes with a specialized driver for this adapter that allows applications to transparently set FD rules [8] whenever a filtering rule is set. This means that whenever an application adds/removes a filtering rule, the PF_RING filtering engine attempts to transparently set the filter

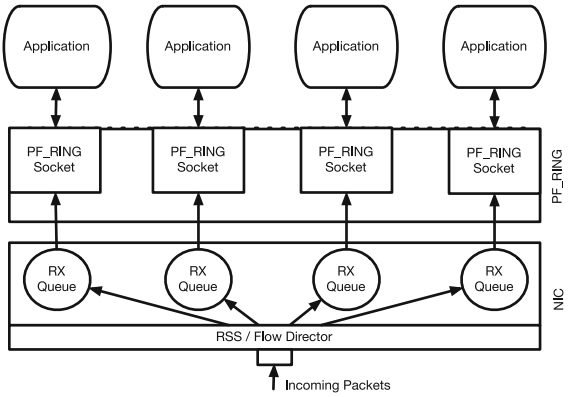


Fig. 19.2 Flow balancing and RX queues in PF_RING

in hardware if the adapter supports it. The result is that unwanted packets are dropped before they hit the driver, hence reducing the amount of packets that need to be processed in software. Captured packets are still filtered in software as the network adapter might not support at all or feature limited hardware filtering capabilities with respect to the filtering rules supported by PF_RING.

The combination of native multi-core/virtual RX support, support of hardware flow filtering/balancing, and in-kernel protocol dissection and analysis, makes the PF_RING framework ideal for the creation of modular and efficient traffic monitoring applications. The following section shows how this technology can be efficiently used for creating service-oriented monitoring applications.

19.4 Using PF_RING for Network Service Monitoring

Over the years, network applications have been constantly updated to implement the latest innovations in security. Although firewalls and IPS (Intrusion Prevention Systems) have been deployed at network borders in order to prevent unwanted communications, it is still necessary to deploy monitoring applications for discovering traffic that circumvents the security policies. This trend is driven, for example, by the use of generic protocols such as HTTP for transporting data and by the spread of technologies for creating network overlays on which freely exchange data. Security threats are also caused by unauthenticated service requests, user service abuse, misbehaving clients and permissive access rules. Web-services technologies and cloud computing are examples of traffic that needs to be carefully inspected in order to implement what is generally called trustworthy Internet. Although most Internet protocols are managed by many security systems already available on the market, it is often necessary to implement fine-grained tools for controlling selected protocol requests and also checking those protocols (e.g. network database communications) that are often not supported by monitoring appliances. Given this, it is necessary to move from packet to

service-oriented monitoring in order to monitor the expected service agreements and usage policies. This requires:

- Going beyond packet header monitoring and inspecting the packet payload in order to analyze the service parameters and validate the responses.
- Computing detailed service metrics in addition to generic metrics such as throughput, latency and used bandwidth.
- Correlating various service requests in order to create a unique service data record rather than several service access requests all related to the same master request.

PF_RING simplifies the process of building service-oriented applications as it provides:

- Filtering, balancing and packet reflection capabilities for implementing simple packet filtering and balancing devices. This allows network administrators to balance the monitoring workload across processor cores which is a key requirement for performing complex resource consuming analysis tasks. To the best of our knowledge, PF_RING is the only open-source framework that can successfully exploit native hardware flow prioritization mechanisms implemented by modern network adapters in the context of traffic monitoring.
- An extensible plugin-based architecture that can be used for inspecting various protocols including Internet (e.g. web and email) and transactional (e.g. database) communications. Developers can focus on dissecting and analyzing packets while leaving the duty of dividing packets per-flow, reordering and discarding duplicates to the framework. The framework is responsible for maintaining per-flow information including protocol metrics and service request parameters.
- Filtering rules for early discarding packets that are not due to be analyzed, and for dissecting selected flows using a specific plugin.
- Correlating flows by exploiting the intra-flow framework mechanisms, for alerting specific plugins whenever a certain flow is created, deleted or updated.

As of today, the PF_RING framework has been successfully used for simplifying the development of complex and yet efficient monitoring software for real-time services [12] and HTTP-based applications. The performance evaluation of the filtering infrastructure can be found in [8], whereas [7] reports the packet capture performance.

19.5 Conclusions

In this paper we showed that network monitoring goals have changed in the past years and that the focus shifted from packet-level analysis to fine-grained service monitoring. The shift requires easy to develop and extend monitoring probes capable for performing complex analysis tasks on modern high-speed networks

by leveraging the latest innovations in computer hardware. High-performance and ease of extensibility can be achieved by creating simple building blocks for handling various low-level activities which allows application developers to focus only on the specific problem they are tackling. From a survey of the various software and hardware technologies available, we came to the conclusion that even if there are several solutions available for tackling specific monitoring problems, there is not a comprehensive framework that can be used as a foundation for developing complex monitoring applications. This has been the driving force for creating PF_RING, an open-source flow analysis framework developed by the authors, which has been successfully used to tackle different service monitoring problems including real-time analysis of multimedia streams and web communications.

References

1. Agarwal, A.: The tile processor: a 64-core multicore for embedded processing. In: Proceedings of HPEC Workshop (2007)
2. Baldi, M., Risso, F.: Using xml for efficient and modular packet processing. In: Proceedings of Globecomm, New York, NY (2005)
3. Bos, H., de Bruijn, W., Cristea, M., Nguyen, T., Portokalidis, G.: Ffpf: fairly fast packet filters. In: OSDI'04: Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation, pp. 24–24. USENIX Association, Berkeley (2004)
4. Brownlee, N., Mills, C., Ruth, G.: Traffic flow measurement: architecture. RFC 2722 (1999)
5. Claise, B.: Specification of the IP flow information export (IPFIX) protocol for the exchange of ip traffic flow information. RFC 5101 (2008)
6. Deri, L.: Improving passive packet capture: beyond device polling. In: SANE 2004: Proceedings of the 2004 System Administration and Networking Conference. USENIX Association (2004)
7. Deri, L., Fusco, F.: Exploiting commodity multi-core systems for network traffic analysis. Technical Report (2010)
8. Deri, L., Gasparakis, J., Waskiewicz, P.J., Fusco, F.: Wire-speed hardware-assisted traffic filtering with mainstream network adapters. In: NEMA '10: Proceedings of the First International Workshop on Network Embedded Management and Applications, page to appear. Niagara Falls, Canada (2010)
9. Dharmapurikar, S., Paxson, V.: Robust TCP stream reassembly in the presence of adversaries. In: SSYM'05: Proceedings of the 14th Conference on USENIX Security Symposium, p. 5. USENIX Association, Berkeley (2005)
10. Donnelly, S.: Dag packet capture performance. http://www.endace.com (2006)
11. Fuji, K.: Jpcap. Homepage http://netresearch.ics.uci.edu/kfujii/jpcap/doc/
12. Fusco, F., Huici, F., Deri, L., Niccolini, S., Ewald, T.: Enabling high-speed and extensible real-time communications monitoring. In: IM'09: Proceedings of the 11th IFIP/IEEE International Symposium on Integrated Network Management, pp. 343–350. IEEE Press, Piscataway (2009)
13. Fusco, F., Stoecklin, M., Vlachos, M.: Net-fli: on-the-fly compression, archiving and indexing of streaming network traffic. In: Proceedings of the 36th International Conference on Very Large Data Bases (VLDB), page to appear (2010)
14. Hyperscan: http://sensorynetworks.com/Products/HyperScan/
15. Intel: 80579 Integrated Processor. http://www.intel.com/design/intarch/ep80579 (2010)
16. Intel: 82599 10 gbe controller datasheet. Rev. 2.3 (2010)

17. Jacobson, V., Leres C., McCanne, S.: Libpcap. Homepage http://www.tcpdump.org
18. McCanne, S., Jacobson, V.: The BSD packet filter: a new architecture for user-level packet capture. In: USENIX'93: Proceedings of the USENIX Winter 1993 Conference, p. 2. USENIX Association, Berkeley (1993)
19. Orebaugh, A., Ramirez, G., Burke, J., Pesce, L.: Wireshark & Ethereal Network Protocol Analyzer Toolkit (Jay Beale's Open Source Security). Syngress Publishing, Rockland (2006)
20. Protocol Plugin Library: http://www.qosmos.com/products/protocol-plugin-library
21. Ptacek, T., Newsham, T., Simpson, H.J.: Insertion, evasion, and denial of service: eluding network intrusion detection. Technical Report, Secure Networks (1998)
22. Wu, Z., Xie, M., Wang, H.: Swift: a fast dynamic packet filter. In: NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, pp. 279–292. USENIX Association, Berkeley (2008)

Chapter 20
Analyzing Telemarketer Behavior in Massive Telecom Data Records

Nico d'Heureuse, Sandra Tartarelli and Saverio Niccolini

Abstract Regulations for the limitation of telemarketing calls are becoming increasingly strict for most industrialized countries. Despite this, telemarketing calls represent a growing trustworthiness issue in today's telephone networks. This calls for systems able to efficiently detect telemarketers, so that adequate countermeasures can be taken. In this paper we first present Voice over IP Secure Application Level firewall (VoIP SEAL), an anomaly detection system with particular focus on telemarketers. VoIP SEAL algorithms are based on measurements performed at the application level. The richness of such information is fundamental for the user characterization. We then provide an overview of the results obtained by using VoIP SEAL to analyze massive sets of telephone data obtained from three European telephone operators. Our results allow quantifying the relevance of telemarketers in today's networks.

Keywords Voice over IP · Telemarketing · Risk assessment

20.1 Introduction

With the decreasing cost for telephony calls, the problem of telephony spam sent by telemarketers constantly increases. Although in some countries telemarketing calls are strongly regulated, they continue to exist nearly everywhere in the world.

N. d'Heureuse (✉)· S. Tartarelli · S. Niccolini
NEC Europe Ltd., NEC Laboratories Europe, Heidelberg, Germany
e-mail: dheureuse@neclab.eu

S. Tartarelli
e-mail: tartarelli@neclab.eu

S. Niccolini
e-mail: niccolini@neclab.eu

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_20,

In Germany, for example, telemarketing calls are illegal without explicit consent of the callee. Nevertheless, the 2009 yearly report of the German Federal Network Agency (Bundesnetzagentur) states that the agency receives approximately 5000 complaints every month regarding unwanted telemarketing calls [1].

Telemarketers misusing the offered flat-rate models of telephone operators, make these tariffs unprofitable for the operators. Furthermore, customers are increasingly annoyed by telemarketers advertising their products or services. Therefore the detection of telemarketers is important for both, the network operator and its customers.

In this paper, we first present Voice over IP Secure Application Level firewall (VoIP SEAL), a system we developed for detecting anomalies and in particular telemarketers in telephone networks. In the literature, several methods were proposed for detecting or preventing telemarketing calls (see Sect. 20.2). Compared to other approaches, VoIP SEAL's basic concept consists of combining several metrics that can be measured at the application level. Users are then classified according to the combination of specific metric values. For instance, telemarketers are expected to be characterized by a large number of outgoing calls, small number of incoming calls, short average call duration, etc.

In the past two years, we extensively tested VoIP SEAL, by carrying out trials with different European operators. The target of the trials was twofold. On one hand we wanted to test VoIP SEAL capabilities of detecting telemarketers. On the other hand, we were interested in quantifying the telemarketers phenomenon in today's telephone networks. Indeed, a major contribution of this paper is that it provides an overview of the telemarketing behavior based on real-world data and also details some basic statistics of the telemarketing traffic.

It is sadly known to all researchers how difficult it is to obtain operational data from operators, thus making the overview provided in this paper especially relevant for the research community.

The remainder of this paper is structured as follows: in Sect. 20.2 we review some related work in the area of telemarketers detection. We then present VoIP SEAL in Sect. 20.3, we give an overview of the data used for the analysis in Sect. 20.4 and we outline the system configuration in Sect. 20.5. Section 20.6 provides an overview of the results we obtained when analyzing massive telephone datasets. Results are then discussed in Sect. 20.7. Finally, Sect. 20.8 concludes our work by pointing out the most relevant contributions of the paper.

20.2 Related Work

The detection of spam calls has been analyzed by several researchers. In [2] the authors propose to combine two different call rates, one considering only a short time windows and one considering a longer period, in order to detect spammers. Instead of detecting the spammers directly, the authors of [3] establish trust relationships between users. The phone calls are used to exchange trust tokens

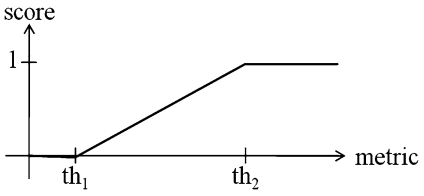


Fig. 20.1 VoIP SEAL's RAMs currently use linear scoring system

between users. As an extension, this trust information can be used to form a social network which allows the derivation of indirect trust relationships. [4] also uses measures of trust and reputation and combines them with the users' presence state to a multi-stage SPAM filtering system. A modified clustering coefficient is presented in [5] which allows the identification of potential spammers in the intrinsic social network formed by phone calls. All the works mentioned so far consider only the call signaling and the information which can be derived from it. In contrast to this, [6] presents a method for the detection of SPAM calls by analyzing the media (i.e., audio) streams.

It is worth noting that all above methods were validated only based on artificially generated calls or on very small sets of real call data.

The general concepts of our framework "VoIP SEAL" have already been described in [7]. In this paper, however, we describe a specific implementation of the presented architecture. Furthermore, we apply our method to massive sets of real-world operational call data and present the results obtained. This is, to the best of our knowledge, the first work which not only proposes a system for telemarketer detection, but which also gives results on massive real-world data.

20.3 NEC's System for Telemarketer and Anomaly Detection

During the last years we developed VoIP SEAL, a flexible system for the detection of anomalies in telephony networks. VoIP SEAL was originally intended for SIP [8] based systems. However, due to its modular design, PSTN networks can be analyzed as well.

The main analysis stage of VoIP SEAL analyzes each call separately, by considering application level semantics. For each call multiple risk assessment modules (RAMs) are invoked. Each RAM performs a certain statistical analysis on the call and returns an anomaly score (s_1 to s_n), ranging from zero (no anomaly detected) to one (very anomalous). For the work presented in this paper, only a subset of the RAMs developed for VoIP SEAL was deployed. For all these RAMs, the anomaly score is obtained by applying a two-threshold linear function (cf. Fig. 20.1) to the RAM specific metric.

In the current implementation, the individual scores are combined into an overall anomaly score for each call, by calculating a weighted sum of the module scores (cf. Fig. 20.2). According to their score, we categorize the calls into three

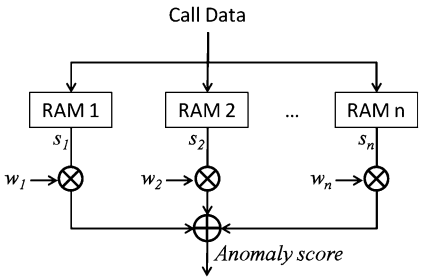


Fig. 20.2 VoIP SEAL's modular architecture

groups using two thresholds t_L and t_H. Calls with a score larger than t_H are labeled "Anomalous", calls with a score less than t_L as "Normal". All other calls (with scores between t_L and t_H) are labeled as "Suspicious".

Similarly, users are categorized as "Normal", "Suspicious" and "Anomalous" depending on the average score of all the calls they initiated. For the user categorization, the same thresholds t_L and t_H as for the call classification are used.

Although, depending on the modules deployed, the system can detect various types of anomalies, our work is currently focused on the detection of telemarketers. Therefore, for the scope of this work, "anomalous" means "very likely to be a telemarketer". The RAMs used for this purpose are listed in Table 20.1 together with a short explanation of the related metric. Note that all RAMs can be configured to start scoring calls only if a user has started at least a minimum number of calls in the considered time window.

20.4 Data Description

We used VoIP SEAL to analyze call data of three European telecoms operators. The calls were provided to us in form of anonymized CDRs (Call Detail Records). For each call, a CDR contains at least the source and destination phone number, the time the call was started, the call duration as well as a cause code or response code.[1] The CDRs we analyzed originate from PSTN as well as VoIP networks. Table 20.2 gives an overview of the four datasets we used for the analysis presented in this paper. For *Operator 2* it shows two different datasets (*Op2-1* and *Op2-2*) with approximately one month between the end of the first dataset and the beginning of the second dataset.

[1] Cause codes or response codes indicate whether a call was established or if an error occurred. In the error case, the code specifies the type of error. Depending on the network type, different codes are used. The most common ones are SIP reply codes [8] and ITU Q.850 cause codes [9]. Our system converts all codes into corresponding SIP reply codes.

Table 20.1 RAMs for telemarketer detection

RAM	METRIC
SOC	No. of concurrently active outgoing calls
FoFiR	Ratio between no. of established outgoing (fanout) and incoming (fanin) calls within a time window. If fanin = 0 then metric is equal to the fanout of the respective time window
URL	Ratio between number of distinct callees (in a time window) and total number of established calls started in a time window
ACD	Ratio between user's average call duration (in a time window) and global average call duration (in a time window)

Table 20.2 Data description

Dataset	Op1	Op2-1	Op2-2	Op3
Data collection time	2008	2009	2009	2009
Network type	VoIP	VoIP	VoIP	PSTN + VoIP
Period	$\approx$ 2 months	$\approx$ 4 weeks	$\approx$ 5 weeks	1 week
Number of calls	$>$100M	$>$20M	$>$30M	$>$100M
Number of users	$>$15M	$>$3M	$>$4M	$>$15M

20.5 System Configuration

The configuration used in the trials aims at identifying telemarketers. In the longer term, we plan to implement a self-configuration mechanism that adjusts the configuration parameters depending on external feedback (e.g, false positive and false negative rates). However, the feedback obtained from the operator so far is still too limited to rely on it for an automatic configuration mechanism. Therefore, in the current version, the system only supports a static configuration. Since the VoIP SEAL RAMs used for telemarketer detection are based on rather "intuitive" metrics, an initial configuration can be worked out based on common sense. We ran initial tests with this configuration, performed some analysis on the anomalous users detected and then adjusted the configuration such that most of the users classified as anomalous by VoIP SEAL show statistics close to what expected for telemarketers. Finally, the RAMs of Table 20.1 were configured to score only "heavy hitters", i.e. users that have a call volume significantly larger than average (we configured this value to $\approx$ 100 calls/day on active days). The same configuration was used for all datasets (cf. Table 20.2) and it is summarized in Table 20.3. Moreover, we set $t_L = 3$ and $t_H = 5$.

20.6 Detection of Anomalous Behavior

Table 20.4 and Fig. 20.3 compare the results of the VoIP SEAL analysis for the four sets of CDRs described in Table 20.2. All operators show a certain percentage

Table 20.3 VoIP SEAL basic configuration for telemarketer detection

RAM	Basic configuration	Weight (w_i)	Time window (h)
SOC	$th_1 = 3$, $th_2 = 5$	2	–
FoFiR	$th_1 = 2$, $th_2 = 10$	2	6
URL	$th_1 = 0.5$, $th_2 = 1$	3	6
ACD	$th_1 = 0.1$, $th_2 = 0.2$	3	24

Table 20.4 Data analysis summary

	Op1 (%)	Op2-1 (%)	Op2-2 (%)	Op3 (%)
Anomalous Calls	0.6	6.4	5.3	7.3
Suspicious Calls	0.6	2.0	1.6	11.7
Anomalous users	$\approx 0.0001\%$	$\approx 0.001\%$	$\approx 0.001\%$	0.02
Suspicious users	0.01	0.01	0.01	0.23
Users with at least one anomalous call	0.05	0.03	0.03	0.5
% of anomal. calls started by anomal. users	11	80	78	67

of anomalous calls within their network. This percentage varies from only 0.6% of all calls for $Op1$ to the significantely higher percentage of 7.3% for $Op3$. These values are especially interesting when compared to the number of anomalous users and the amount of calls they started. The percentage of users classified as anomalous is extremely low for $Op1$ ($\approx 0.0001\%$) and $Op2$ ($\approx 0.001\%$). For $Op3$ such percentage amounts instead to (0.02%), which is anyway a relatively low percentage. The last line in Table 20.4 shows, however, that this small percentage of users is responsible for most of the anomalous calls in the case of $Op2$ and $Op3$. This means that in these networks only very few users are responsible for a significant amount of anomalous calls. Thus, the ability of identifying such users is extremely relevant for increasing the trustworthiness of these networks.

Different is the case of $Op1$, for which the anomalous users are responsible for only 11% of the anomalous calls. The remaining anomalous calls are then started by users that usually behave "normally" and only show some sporadic anomalous pattern (such users are not classified as anomalous on average). Note that the latter type of users are probably not telemarketers. Indeed, from the feedback we got from the operators, $Op1$ seems to be the one that experiences the least serious problems with telemarketers (at least by the time the trial was carried out).

Next, we investigated the distribution of the anomalous calls over time. Fig. 20.4 shows the number of calls per class (normal/suspicious/anomalous) of the $Op2-2$ dataset for each day. One can clearly observe that the overall call volume follows the typical weekly behavior, with more calls being initiated on weekdays than on weekends. However, a non-negligible amount of anomalous calls is hidden within this, at first glance, normal behavior. Furthermore, in this trace—and in all other traces we analyzed—the relative percentage of anomalous and suspicious calls is higher on weekdays than on weekends.

To summarize, this section shows that, in all networks analyzed, the percentage of anomalous calls is significantly high, especially during weekdays. Moreover, for

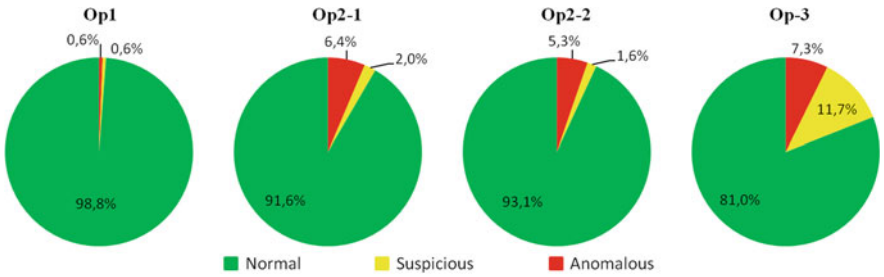


Fig. 20.3 Call classification

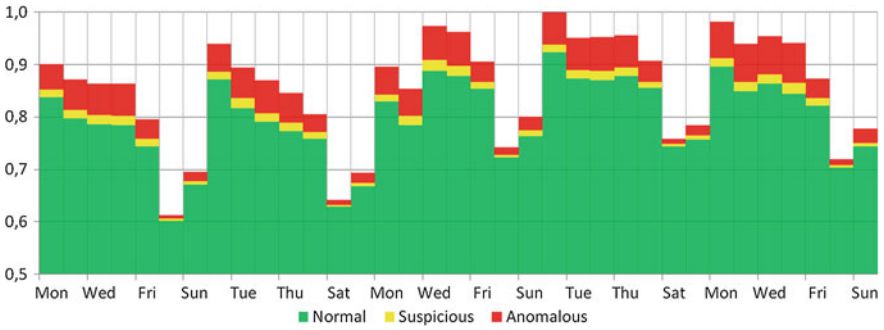


Fig. 20.4 *OP2-2*: Call classification per day (normalized) over five weeks

two of the operators, few users are responsible for most of the anomalous calls. Therefore, identifying such limited set of anomalous users allows managing a high portion of the anomalous traffic.

20.7 Analysis of Anomalous Traffic

The target of this section is twofold. First, it aims at providing some validation for the reliability of the results presented in the previous ion. Second, it gives a quantitative analysis of several traffic statistics for different operator networks. Given the common operators' reluctance to disclose their operational data, despite being it anonymized, this is in itself a considerable contribution.

A rigorous validation of the results in the previous section would require the availability of a "*ground truth*". This means that operators should de-anonymize and trace back all users in order to point out false positives (i.e., normal users wrongly classified as anomalous) or false negatives (i.e., anomalous users not detected by VoIP SEAL). This approach is clearly not feasible for the operators. Nevertheless, we tried to validate our results using a combination of traffic analysis and a very limited "*groundtruth*" we obtained from one of the operators.

Table 20.5 Data analysis summary

	Op1	Op2-1	Op2-2	Op3
Average calls per day per user (all)	0.49	0.42	0.32	5.55
Average calls per day per user (top50)	336.27	1027.25	662.58	4308.31
Average call duration (all) (s)	269	332	345	420
Average call duration (top50) (s)	68	53	53	60
Avg. establishment Rate (all) (%)	57	65	70	73
Avg. establishment Rate (top50) (%)	59	32	38	55
Avg. No. of calls per callee, established calls (all)	3.40	2.70	3.08	2.32
Avg. No. of calls per callee, established calls (top50)	359.45	1.66	1.77	1.82
Avg. No. of calls per callee, all calls (all)	4.18	3.28	3.57	3.36
Avg. No. of calls per callee, all calls (top50)	372.58	3.00	2.90	2.09

This section presents some results obtained during the validation process. The first analysis we performed was to compare some basic metrics of anomalous users against average values measured on the entire traffic (see Table 20.5). We selected those metrics that we expected to be significantly different between telemarketers and normal users. We arbitrarily decided to evaluate statistics for the top 50 anomalous users (i.e., the 50 users with the highest average anomaly score). The reason was mainly to concentrate on a subset of users that have relatively similar anomaly score values. Table 20.5 shows different averaged metrics for the top 50 as well as for all users of each dataset. Note that the *average call duration* is used directly by a RAM for scoring calls. Also, all RAMs give a positive score only if a configurable minimum number of calls per time window is reached (see Sect. 20.6) Therefore it is natural to observe a large difference between average values for all users and for the top 50 users with respect to the average call duration and the number of calls initiated on average each day.

More insightful is instead the information we can derive from the remaining two metrics, i.e., *establishment rate* and the number of *calls* a user places *per callee* over the entire observation period, which are not directly used for scoring calls. One would expect these metrics to differ for telemarketers and regular users, i.e., home users or companies. Indeed, there are several factors that may contribute to reduce the establishment rate of telemarketers. First, telemarketers do not know the habits of the callee, so they have a smaller probability of reaching the callee at a time when he/she is available. A second factor contributing to a smaller establishment rate, may be due to the use of predictive dialers [10]. Telemarketers typically use this software to avoid agents being idle for too long. Predictive dialers usually start many more calls than those that can be handled by the agents. If at a certain point all agents are busy, calls are simply dropped by the automatic dialer. A third factor may be related to the use of number scanning, resulting in a relatively high percentage of calls being directed towards not existent numbers. Indeed, Table 20.5 shows a much lower establishment rate for the top 50 users compared to the overall average.

Also for the number of calls started by one user and directed to the same recipient, one would expect a telemarketer typically not to call again the same

Table 20.6 Reference values for telemarketers' behavior in *Op2-1* trace

	Avg	Stdev	Min	Max
Average calls per day per user	1104	744	115	4198
Average call duration (s)	41	14	27	67
Establishment Rate (%)	21	10	8	39
Calls per callee, established calls	1.6	0.3	1.2	2.1
Calls per callee, all calls	4.1	1.9	1.8	7.9

user, if that user has already been contacted. This is different for normal users, who instead tend to call the same set of users. Indeed, also in this case, the values obtained for the top 50 users are significantly different from those of all users. This analysis represents itself a first validation of the reliability of VoIP SEAL for the identification of telemarketers.

It is worth noting that values in Table 20.5 are significantly different for *Op*1 with respect to the other CDR sets. The explanation is manyfold. First, the VoIP network of *Op*1 is still in an initial phase compared to the other operators. As already mentioned, we know also from the operator that they have significantly less problems with telemarketers compared to *Op*2 and *Op*3. Besides, the number of anomalous users in *Op*1 is so small that the top 50 users by anomalous score that contributed to the values in Table 20.5 actually included several "suspicious" users. Among these suspicious users, we observed for instance some who started many calls, always directed to the same callee or to a small number of callees. These users are clearly also "anomalous" in the general sense (they may be either test numbers or even denial of service attacks), but they are not telemarketers.

As mentioned previously, one of the operators gave us feedback about a small set of telemarketers (seven in total) identified by VoIP SEAL for the *Op*2−1 trace. So we were able to build at least a limited "*ground truth*" on a set of users that we knew were telemarketers. Table 20.6 reports the related measures.

By comparing results in Table 20.5 and those of the "confirmed" telemarketers in Table 20.6, we can see that values for the top 50 anomalous users of *Op*2−1 identified by VoIP SEAL fall within the minimum and maximum values of the "confirmed" telemarketers for the same dataset. This is an additional indication of the reliability of VoIP SEAL classification for telemarketers.

Given the metrics used by VoIP SEAL to detect telemarketers, we wanted to check whether also medium or large companies that only have one identifier for outgoing calls may show a behavior similar to that of telemarketers. Unfortunately, we did not have any indication from the operators about which identities were associated to companies. We therefore decided to take statistics of the telephony logs of our company's network, as an example of SME with approximately 100 employees. Clearly, results cannot be generalized; nevertheless they provide at least one example of a user being very active without being malicious. In our network, we measured for outgoing calls around 150 calls/day (only weekdays), establishment rate of about 69%, and an average call duration amounted to about 400 s. The latter two values are very close the overall averages of at least *Op*2 and

*Op*3 traffic, confirming the fact that not malicious users should have a different behavior compared to telemarketers.

20.8 Conclusions

This paper first presents VoIP SEAL, our system for telemarketers detection in telephone networks. Afterwards, it provides a summary of the results obtained during trials we carried out with three European operators. The aim of the trials was twofold. First, we wanted to test VoIP SEAL's capabilities of detecting telemarketers. The comparison of VoIP SEAL performance against selected techniques presented in the literature is work in progress and was not included in the paper. Second, we intended to quantify the trustworthiness issue raised by telemarketers in today's networks. Results showed that telemarketers are definitely a not negligible phenomenon. Besides, for two of the three operators considered, users classified by VoIP SEAL as telemarketers were responsible for a large percentage of all observed anomalous calls. In these cases, the detection of such users is especially relevant for the operator and its customers. Additionally, the paper provides useful information about several statistics taken on real telephone traffic. For researchers it is always very difficult to have access to real-world data; therefore some of the statistics reported in this paper represent a useful reference also for related research activities. We are indeed not aware of any other work analyzing and comparing statistics of massive amounts of real telephony traffic of different operators' networks.

Acknowledgements This work was partially supported by DEMONS, a research project supported by the European Commission under its 7th Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DEMONS project or the European Commission.

References

1. Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen Presse und Öffentlichkeitsarbeit, Jahresbericht der Bundesnetzagentur 2009. http://www.bundesnetzagentur.de/cae/servlet/contentblob/152206/publicationFile/6683/Jahresbericht2009Id18409pdf.pdf (2009)
2. Shin, D., Ahn, J., Shim, C.: Progressive multi gray-leveling: a voice spam protection algorithm. IEEE Netw. Mag. **20**(5), 18–24 (2006)
3. Balasubramaniyan, V.A., Ahamad, M., Park, H.: CallRank: combating SPIT Using Call Duration, Social Networks and Global Reputation. In: CEAS 2007 Fourth Conference on Email and AntiSpam (2007)
4. Kolan, P., Dantu, R.: Socio-technical defense against voice spamming. ACM Trans. Auton. Adapt. Syst. **2**(1), 2 (2007)

5. Kusumoto, T., Chen, E.Y., Itoh, M.: Using call patterns to detect unwanted communication callers. Applications and the Internet, IEEE/IPSJ International Symposium on, vol. 0, pp. 64–70 (2009)
6. Rebahi, Y., Ehlert, S., Bergmann, A.: A SPIT detection mechanism based on audio analysis. In: Proceedings of 4th International Mobile Multimedia Communications Conference, July (2008)
7. Quittek, J., Niccolini, S., Tartarelli, S., Schlegel, R.: On spam over internet telephony (SPIT) prevention. IEEE Commun. Mag. **46**, 80–86 (2008)
8. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393
9. ITU-T, Usage of cause and location in the Digital Subscriber Signalling System No. 1 and the Signalling System No. 7 ISDN User Part. ITU-T Recommendation Q.850, May 1998
10. Wikipedia, Predictive dialer. http://www.en.wikipedia.org/wiki/Predictive_dialer

Chapter 21
SIP Overload Control: Where are We Today?

Dorgham Sisalem

Abstract Like any other Internet-based service VoIP servers can get overloaded. This overload can result from a denial of service attack, flash crowd scenario or hardware and software problems on the VoIP server itself. To be able to properly react to the overload the VoIP servers will have to monitor the incoming traffic and identify the causes of the overload. Depending on the cause of the overload and the overloaded resource different overload control mechanisms will have to be deployed. Without such mechanisms, VoIP services can fail and can hence not be trusted as replacement to PSTN services. This paper looks into the issues that need to be solved with regard to controlling the load at SIP servers, the different possibilities for controlling the load and some of the still open issues.

Keywords Overload control · SIP · Voice over IP · Monitoring

21.1 Introduction and Motivation

VoIP servers whether proxies, application servers or session border controllers constitute the core components of any VoIP infrastructure. These servers are responsible for processing and routing VoIP signaling traffic and supporting various advanced services such as call forwarding or voicemail. In order to ensure a highly available VoIP service, these servers will have to continue performing their tasks even under high load situations.

D. Sisalem (✉)
Tekelec, Berlin, Germany
e-mail: dorgham.sisalem@tekelec.com

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_21,

Overload can occur as the result of a denial of service attack or from software or hardware problems on the VoIP server itself. Like any other Internet-based service VoIP servers can be the target of denial of service attacks. While operators will certainly deploy various mechanisms to prevent such attacks, detecting and preventing DoS attacks is not a trivial task. Attacks can be disguised as legitimate VoIP traffic so distinguishing between a denial of service attack or a sudden surge in traffic due to some event is not always possible. Hence, VoIP servers need to incorporate mechanisms that monitor the load and the incoming traffic, identify the overloaded resources and cause of the overload and react in a manner that will prevent a complete service interruption. Without such mechanisms, VoIP systems will fail under overload situations. Work done in [1] suggests that overload situations not only reduce the performance of a SIP server but can finally lead to a complete standstill of the entire VoIP service. The unavailability of the VoIP service due to overload would not only be inconvenient to the subscribers but would also tarnish the reputation and the trustworthiness of the provider. Actually with a reputation of being insecure and unreliable, the VoIP technology would suffer as a whole with dramatic economic consequences and will reduce the trustworthiness of VoIP as a replacement of PSTN.

In designing overload control mechanisms for VoIP servers, monitoring and identifying the cause of the overload plays a major role in deciding the system's reaction to the overload situation. On the one hand, when overload is caused for example by a denial of service attack, it would be useless to redirect the incoming traffic to another server, as this would only result in overloading that server as well. On the other hand, redirection would be an appropriate option when the overload situation is caused by software, hardware failures or unbalanced traffic load.

Currently, a lot of the offered VoIP services are based on the session initiation protocol (SIP), [2]. SIP, however, does not offer sufficient mechanisms for handling overload situations. Hence, in this paper, we look at the available solutions, categorize the different approaches and list some of the open issues.

In Sect. 21.2 we take a brief look at possible causes of overload and the current status of congestion control for SIP. In Sect. 21.3 the different design aspects of a SIP overload control scheme are discussed. For each point the different alternatives are presented and the pros and cons of each alternative are highlighted.

21.2 Overload Control in the SIP Specifications

In its simplest form a SIP-based VoIP service consists of user agents (UA), proxies and registrar servers. The UA can be the VoIP application used by the user, e.g., the VoIP phone or software application, a VoIP gateway which enables VoIP users to communicate with users in the public switched network (PSTN) or an application server, e.g., multi-party conferencing server or a voicemail server. User agents keep state information about each active call that they are processing for the entire duration of the call.

The registrar server maintains a location database that binds the users' VoIP addresses to their current IP addresses.

The proxy provides the routing logic of the VoIP service. When a proxy receives SIP requests from user agents or other proxies it also conducts service specific logic, such as checking the user's profile and whether the user is allowed to use the requested services. The proxy then either forwards the request to another proxy or to another user agent or rejects the request by sending a negative response.

A SIP proxy acts in either statefull or stateless mode. In the statefull mode, the proxy forwards incoming requests to their destination and keeps state information about each forwarded request until either a response is received for this request or a timer expires. If the proxy did not receive a response after some time, it will resend the request. In the stateless mode, the proxy would forward the request without maintaining any state information. In this case the user agents would be responsible for retransmitting the request if no responses were received. As the statefull behavior reduces the load on the user agents and provides the service provider with greater session control possibilities, e.g., forwarding the request to another destination if the first one did not reply, statefull SIP proxies are usually used by VoIP providers.

With regard to the SIP messages we distinguish between requests and responses. A request indicate the user's wish to start a session (INVITE request) or terminate a session (BYE request). We further distinguish between session initiating requests and in-dialog requests. The INVITE request used to establish a session between two users is a session initiating request. The BYE sent for terminating this session would be an in-dialog request. Responses can either be final or provisional. Final responses can indicate that a request was successfully received and processed by the destination. Alternatively, a final response can indicate that the request could not be processed by the destination or by some proxy in between or that the session could not be established for some reason. Provisional responses indicate that the session establishment is in progress, e.g, the destination phone is ringing but the user did not pickup the phone yet.

In the context of this paper we will use the term SIP server to indicate any SIP component that is expected to handle many calls in parallel. This includes SIP proxies, application servers, conferencing servers or PSTN gateways. As SIP proxies constitute the core components of a VoIP service and will have to handle signaling traffic arriving from thousands if not millions of user agents more attention and details will however be given to proxies.

A SIP server can become overloaded due to various reasons such as:

- *Denial of service (DoS) attack*. DoS attacks on a SIP server can take different forms and target either the memory consumption of the server or the CPU or both [3].
  - *Flooding attacks*. With these kinds of attacks, an attacker generates a large number of SIP requests. Even if these requests end up being dropped by the server, the server will first have to parse and process them before deciding to either forward, reject or drop them. Depending on the number of generated

requests, such attacks can misuse a large portion of the CPU available to the server and reduce the amount of CPU available for processing valid requests.
  - *Memory attacks*. This is a more intelligent kind of attack in which the attacker sends valid SIP requests that are forwarded by the server to destinations that do not answer properly. With each forwarded request the server will maintain some transaction state. If the destination of these requests does not answer at all, then the server will keep on trying for some time, usually 32 seconds, the so called Timer B in [2], before it can delete the state information. If the destination answers with a provisional response but not with a final one, then the server will have to keep the transaction state for at least 3 minutes, the so called Timer C in [2].

- *Flash crowds*. Flash crowd scenarios describe a sudden increase in the number of phone calls in the network. An example for this is when thousands of users want to vote on a TV show. This sudden increase in the number of calls will result in an increase in the required CPU and memory at the server.
- *Unintended traffic*. Software errors or configuration mistakes can cause one or more user agents or SIP proxies to send unintentionally multiples of the amount of the traffic they usually generate. Even though the excess traffic is not generated with malicious intent it is just as useless as DoS traffic and can just as well cause an overload situation.
- *Software errors*. Software errors include memory leak problems or infinite loops. Memory leak would deplete the available memory in a similar manner as a memory DoS attack. An infinite loop could block the server from serving SIP requests.

The most straightforward approach for avoiding overload is to ensure that the available processing resources of a SIP component are sufficient for handling SIP traffic arriving at the speed of the link connecting this component to the Internet. With modern access lines reaching gigabit speeds, provisioning the VoIP infrastructure of a provider to support such an amount of traffic, which is most likely several times the normal traffic can easily become rather expensive.

The SIP specifications do not provide much guidance on how to react to overload conditions. [2] indicates that a server that is not capable of serving new requests, e.g., because it is overloaded, could reject incoming messages by sending a 503 *Service unavailable* response back to the sender of the request. This signals to the sender that it should try forwarding the rejected request to another proxy and not to use the overloaded proxy for some time. Further, the 503 response includes a *Retry-After* header indicating the period of time, during which the overloaded server should not be contacted. While this reduces the load on the overloaded proxy, it results in directing the traffic, which has caused the overload to another proxy, which might then get overloaded itself. Figure 21.1 depicts a scenario in which a load balancer distributes the traffic to two different proxies. In the case of a DoS attack it is most likely that all the SIP servers in a SIP cluster will be affected and will be overloaded at the same time. When the first server replies with a 503, the load balancer will forward the traffic destined to that server to the other server. With the additional traffic this server will become overloaded as well and will

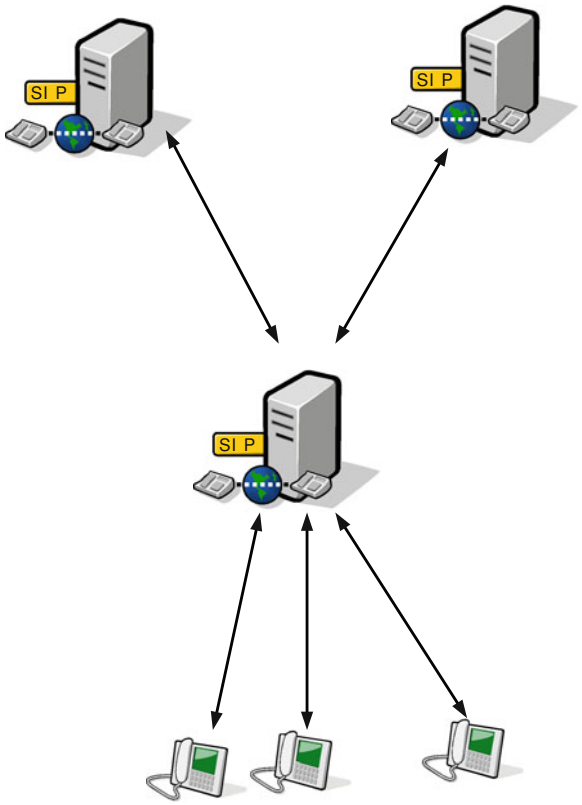


Fig. 21.1 Load distribution and usage of 503 replies

issue 503 replies. Shifting the traffic from one server to another has only made the situation worse for this server. This shifting of traffic can also lead to an on-off behavior. That is, consider the case when an attacker is generating traffic that is causing both servers to run at 100% of their capacity. When one of them issues a 503 response, the traffic destined to it will be forwarded to the other, which will now receive traffic at 200% of its capacity. This server will, hence, issue a 503 response. In case the *Retry-After* value of the first server expires before that of the second server, then that server will suddenly receive traffic at 200% of its capacity and will reject the traffic with another 503. This on-off behavior can actually even lead to a lower average throughput making the 503 approach not optimal for the cases in which a SIP component receives SIP messages from only a small number of other SIP instances. In case a SIP server receives requests from a large number of user agents, then the 503 approach can work much more efficiently as only the user agents that receive the 503 response will try another destination. Further, the on-off behavior would not be observed in this case as spreading out the 503 among the clients has the effect of providing the overloaded SIP instance with more fine-grained controls on the amount of work it receives. Naturally, if the senders are

malicious and do not respect the *Retry-After* header using 503 will not be sufficient for protecting the server from getting overloaded.

As guidelines for providing mechanisms for solving the overload control issue, in [4] the authors discuss the requirements an overload control scheme should fulfill. Among these requirements, is that an overload control scheme should enable a server to support a meaningful throughput under all circumstances, should prevent forwarding traffic to other servers that might be overloaded themselves and should work even if not all servers in the network support it.

Hilt describes in [5] the needed extensions for SIP in order to exchange overload related information between neighboring servers. [6] provides a general overview of the design considerations for overload control schemes and their advantages and some of the issues that need to be considered when designing a new overload control algorithm such as which network topologies to consider and fairness.

21.3 Design Considerations for SIP Overload Mechanisms

When designing a mechanism for protecting a SIP server from overload, the designer needs to answer the following questions:

- *Monitoring of overload indicators*. What information should be monitored and used as the indication of the load status of the server?
- *Algorithm*. Once overload was observed what is the best approach for reducing the overload and what issues should be considered in this context?
- *Reaction*. How should a server reduce the amount of traffic once the overload control algorithm decides that the load must be reduced?
- *Model*. Should the mechanism work in isolation on the server or can some form of cooperation between the SIP servers in the network be assumed? In the standalone model, the SIP server monitors its own overload status and once overload is identified, an overload control algorithm is executed and the server reacts to the excess traffic. In the cooperative model, once overload is identified, the SIP server informs its neighbors about the overload status and requests them to reduce the amount of traffic they are sending to it.

21.3.1 Overload Monitoring

A system is considered to be overloaded if some pre-defined thresholds are exceeded. These thresholds might relate to either the resources used by the system for conducting its task or some values that define the capabilities of the system. By monitoring these resources a system can decide whether some reaction is needed or not. The reaction to the overload will depend on the cause of the overload. The reasons for overload can be either natural such as flash crowds or

due to the failures of some servers in a cluster, accidental as is the case with software or configuration errors or malicious as is the case with DoS attacks. Hence, as part of the monitoring process, the servers will also need to determine the type and nature of the incoming traffic.

A simple value that defines the capability of a system is the number of supported active calls. As each SIP system can maximally support a predefined number of active calls, by monitoring the number of parallel calls a server can easily assess its status.

The authors in [7] present a number of overload control mechanisms that use the number of active calls in the system as the overload indication. The number of active calls is determined as the difference between counted INVITE and BYE requests over a measurement interval. The results shown in [7] suggest that the discussed control mechanisms can keep the SIP systems running at their maximum capacity. However, the general applicability of the presented mechanisms requires more investigation. Determining the maximum number of supported calls at a SIP system is not a trivial task as this will depend on the call model—calls conducted according to the IMS SIP model [8] use more messages than plain SIP as defined in [2],—the call scenario and sizes of the SIP messages and so on. Besides the difficulty in estimating the maximum capacity of a system, it is unclear how the system will react when it is attacked. A DoS attack with a high number of INVITE requests that are never terminated will mean that the number of active calls in the system will be rather high but the system will not be really overloaded as it does not have to serve these calls after processing the initial INVITE. Further, a DoS attack consisting of a flood of in-dialog requests will consume resources but will not be counted for as part of the active dialogs. Hence, when monitoring the number of active calls, the server will need to distinguish between successful and failed calls, calls that were actually completed and ones that just consume resources and are never completed and messages that are not part of a running call or do not belong to any active call.

In order to be able to receive and process a SIP request, a SIP server requires three types of resources:

- *Network buffer*. Before handing a request to the process that is running the SIP logic at the server messages are queued in the UDP/TCP receive buffers. When a system is overloaded, i.e., the rate at which SIP messages arrive is higher than the rate at which the SIP process reads out messages from the buffer, the buffer will fill up and the incoming messages will be dropped. Setting the buffer size to a too small value could lead to the loss of messages during bursts of traffic. Setting the buffer to a too large value, in order to compensate for the effects of bursts for example, could, however, mean that some messages will not be processed in time and the sender will retransmit the request.
- *Memory*. A SIP server needs to copy each incoming request in its internal buffers to be able to process the message. The amount of buffered data and the time period the server is supposed to keep the buffered data varies depending on whether the server is working in a stateful or stateless mode. In any case, the

server will at least need to maintain the buffered data while contacting another entity such as a DNS server or a database for example. Depending on the message type, the number of headers and the body of the message, the size of a SIP message might range from a few hundreds of bytes up to a few thousands. The buffering duration will also depend on the message type and might range from a few seconds to a few minutes.

- *CPU*. After receiving a SIP message, the SIP server needs to parse the message, do some processing and forward the message. Depending on the content and type of the message and server policies the actual amount of CPU resources might vary.

Monitoring the buffer size as an indication of the overload is problematic. This will require frequent reading of the buffer size. This is acceptable when UDP is used as the transport protocol as there would be only one receive buffer. When TCP is used, the SIP system might need to query the receive buffer of thousands of TCP connection which will consume a considerable amount of the CPU resources of the SIP system. Hence, schemes designed for overload control for SIP over TCP that use the buffer occupancy as an indication of overload already note that such schemes are only applicable for scenarios in which the number of communicating components is limited to a few, see [9]. Further, the socket receive buffers tend to be either full or empty. That is, while the system can serve the incoming requests the receive buffer will be empty. Once the SIP system reaches its capacity it will no longer be able to serve all the incoming requests and the receive buffer will be filled up more or less instantly. Therefore, the SIP system should react to overload before the receive buffer becomes full as by then it would be already too late. Finally, the SIP system has no knowledge of content of the buffer. Hence, it does not know whether the queued messages are responses, INVITE requests or other requests and cannot decide on the appropriate reaction.

The authors in [10] monitor the memory usage of the SIP process and use the the memory occupation level as the indication of overload. The memory used by the SIP process itself can be seen as an extension of the receiver buffer memory except that there is more of it and the SIP process knows the content of saved information. This knowledge can enable the SIP system to identify what kind of calls and transactions are active and whether the maximum limit will be reached soon. However, setting an appropriate value for the maximum memory to support is just as problematic as determining the number of maximum active calls as different types of requests require different amounts of memory.

The authors in [10] also discuss the possibility of using an averaged value of the CPU usage as an indication of overload. That is, reaction to overload is triggered once the CPU usage value reaches a certain threshold. This approach has the advantage that it does not really need to know the maximum call processing capacity of the SIP system as is the case when using memory or number of active calls. However, CPU usage values can be rather oscillatory in behavior. Most used SIP systems use multi tasking operating systems in which the CPU is used by multiple processes. Hence, a sudden increase in the CPU usage value could imply an

increase in the SIP processing but could also stem from a background process doing some maintenance tasks. This can lead to an oscillatory overload control behavior.

21.3.2 Overload Control Algorithms

Designing congestion control algorithms was and still is one of the major Ph.D. topics in the area of networking and communication. Over the past 40 years or so hundreds if not thousands of papers were published describing all kinds of overload control schemes. While these algorithms differ in their details, they all share the same concept: identify some value as the overload indication and react to the congestion in some manner.

In theory, most of the overload control schemes published for all kinds of IP-based service could be used for SIP servers as well. This is actually the approach taken by a lot of the papers discussed here. However, congestion control schemes usually assume that all incoming events require the same processing resources. Thereby, reducing the number of processed events by a certain value will release occupied system resources by a predictable value. This is, however, not so straightforward with SIP as different SIP messages and events can require different resources. For example:

- *Requests versus responses*. Successfully processing a response means that the resources that were occupied by a transaction can be freed. On the other hand accepting a new request will occupy some new resources. Hence, processing of responses should have a higher priority than the processing of requests.
- *Call initiation versus in-dialog requests*. Accepting a new call means that the server will have to also be ready for all other in-dialog requests that will come as part of this new call, e.g., BYE or updates of the session. On the other hand, accepting a BYE would mean that a call can be successfully terminated and the resources blocked by the call can be freed. From a user perspective, a successful session termination is also more important than a successful session initiation. As a BYE request usually indicates to the billing system that is should stop accounting a certain call, loosing a BYE could lead to open-ended call records and angry customers.
- *IMS versus IETF call models*. The call models defined for IETF based SIP are simpler than those defined for IMS calls and require fewer messages. If a server is handling both types of calls then accepting a call that is generated by an IMS subscriber will result in overall higher load than if a SIP call was accepted.
- *Call priority*. Similar to the PSTN, a call by some governmental agencies or an emergency call must have higher priority than normal calls. There can also be a distinction between business subscribers and others.
- *User profile*. The amount of resources that will be consumed by an incoming call might differ depending on the complexity of the user's profile, e.g., whether call forwarding or call forking for example is set-up.

Thereby, when designing a congestion control scheme for SIP there are various levels of priorities that should be taken into account. The type and importance of these priorities will depend very much on the server type and deployment scenario. An optimal overload control scheme would enable a server to achieve a maximal throughput by first rejecting or dropping the events with the lowest priority and highest resource consumption. However, this can only be achieved by first parsing the incoming messages which already consumes some resources. Hence, some balance must be found between spending resources for parsing incoming messages and risking the loss of high priority events, which in the worst case can lead to the loss of further resources. For example, blindly rejecting a BYE for a certain call would mean that the server would have to maintain all resources that were blocked by the call.

21.3.3 Reaction to Overload

As discussed in Sect. 21.3.1, a SIP server is said to be overloaded if one or more of its resources is having a value above some maximal limits. Going above these limits can destabilize the system and even lead to complete failure, see [11]. Hence, the goal of any overload control scheme is to reduce the usage of these resources. Reducing the load on the SIP server can be realized in one of the following ways:

- *Drop excess traffic*. In this case all the available resources are used for processing incoming traffic and requests that can not be processed are dropped. As SIP user agents retransmit requests for which they did not receive a response, dropping requests will actually even lead to more traffic in the network. That is, when a SIP client using UDP does not receive a reply after a period of time (T1) it retransmits the request. If no answer was received after 2T1 s then the request is retransmitted again and so on. Thereby a dropped request can cause the retransmission of up to 10 requests. To demonstrate the effects of dropping requests, we conducted a test in which a SIP proxy was overloaded by sending more traffic than it can process. The proxy had a processing rate of 1 request per second and we generated a continuous stream of requests at rates of 1500, 2000 and 2500 requests. All requests arriving on top of the proxy's processing rate were dropped. The results presented in Fig. 21.2 show that even though the requests are generated at a rate of only 1,500 requests per second, due to the retransmissions the proxy will end up having to deal with 10,000 requests per second.
- *Reject excess traffic with* 503. When a server is overloaded it can reject incoming requests with 503 to indicate when the sender can retry using this server. The overloaded server would then not be used by the sender for the retry period indicated in the 503 message. As this can lead to an oscillatory behavior as described in Sect. 21.2 this should only be used for rejecting traffic from end systems and not from other SIP servers.

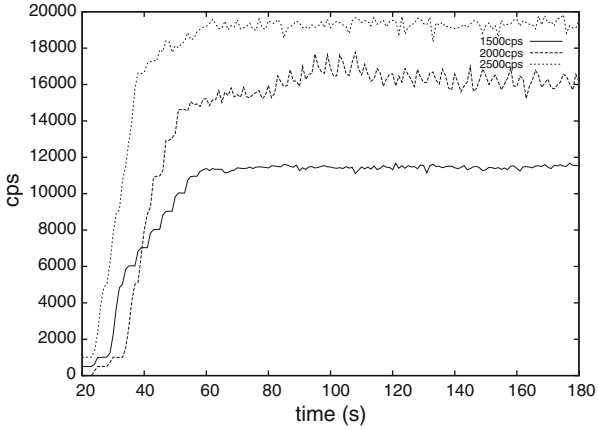


Fig. 21.2 Effects of retransmission on amplification of traffic

- *Reject excess traffic with 5xx*. Instead of dropping excess traffic or requesting not to receive any traffic for some period of time, an overloaded server can reject incoming requests by sending a 500 response for example. This is based on the assumption that rejecting a request is less costly in terms of processing resources than accepting and forwarding it. To test this assumption we ran a number of measurements in which INVITE requests were sent to a SIP server. The SIP server was configured to either forward, reject or drop all incoming requests. Fig. 21.3 depicts the CPU resources used in each case. From Fig. 21.3 we can observe that the amount of resources used for forwarding requests is much higher than that for rejecting them. INVITE messages that are successfully forwarded will be followed by another two requests at least, namely ACK and BYE. Thereby, rejecting an INVITE message will not only save the resources needed for processing the INVITE message itself but would also save the resources that would have been needed for processing other in-dialog requests as well. Rejecting a request is also more resource intensive than just dropping it as it requires processing of the request first. However, when considering that the amount of traffic generated through the retransmissions it would still be preferable.
- *Explicit congestion information*. In this case the overloaded server would actually indicate in its replies its overload status and indicate either the explicit transmission rate the sender should be using or by which percentage the sender should reduce its transmission rate. The authors in [5] describe different possibilities for exchanging information between neighboring SIP systems. These include adding the information in a SIP header or using the SUBSCRIBE/NOTIFY mechanism of SIP [12].

Sending explicit congestion information does not reduce the load at the SIP system by itself. Hence, this approach assumes a cooperative model in which the receivers of the congestion information use this information to actively reduce the

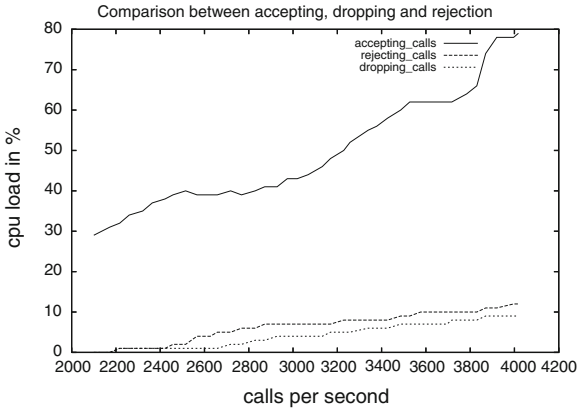


Fig. 21.3 CPU costs of forwarding, rejecting and dropping requests

amount of traffic they send to the overloaded system. Rejecting or dropping traffic reduces the load at the SIP system as it decreases the number of active calls the system has to deal with. By monitoring the number of failed or rejected calls a SIP server can also have some indication of the load status of its neighbors, see. While not as accurate a the explicit congestion information it can be helpful in reducing the overload in the network and does not require the sending and receiving SIP systems to agree on the format used for exchanging congestion information [13].

21.3.4 Standalone or Cooperative Control Model

In general one can distinguish between standalone and cooperative overload mechanisms. In the standalone approach, an overload control mechanism is implemented at the SIP server to be protected. This server monitors its own resources, e.g., memory, CPU, buffer and capacity. Based on the monitored resources the server will recognize when it starts to become overloaded and will have to deal with the incoming traffic by either rejecting new calls or dropping them.

The other approach for overload control is more of a cooperative process. In this scenario the overloaded server would regulate the amount of traffic it is receiving from its neighbors by informing them about its current load. The neighboring servers will then adapt the amount of traffic that they are sending to the overloaded server. In case they have to reduce the amount of traffic they want to send to the overloaded server then they could also inform their neighbors to send less traffic to them.

Both approaches have their pros and cons. The standalone approach can be deployed without having to rely on other SIP components in the network to also support overload handling. It also does not require the standardization of how to exchange status information. This makes this approach the ideal one to start with. However, such a mechanism does not cause the overall load of the SIP network to go

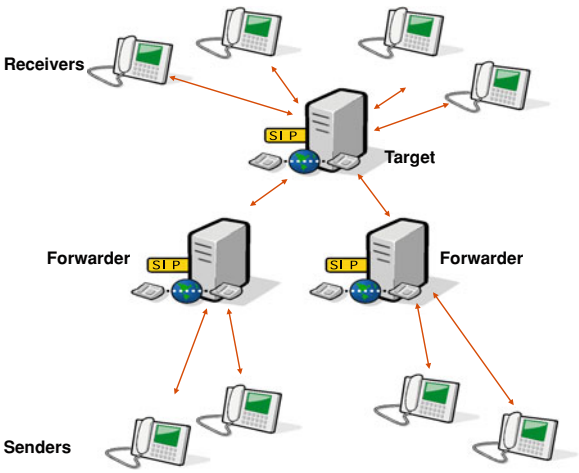


Fig. 21.4 Multi hop communication

down and could lead to situations in which the overloaded SIP system will use all of its resources to just reject or drop incoming requests and will stop serving any calls.

The cooperative approach can adapt the number of calls in the network to the actually available resources and could push the overload to the borders of the network. In this way, excess calls will be prevented from even reaching the overloaded servers and access points can consider using non-overloaded paths for establishing the calls. On the negative side, a server that would ignore the feedback information would still cause overload and packet drops. Actually not deploying the overload control mechanism would give the non-conforming system an unfair advantage.

Another issue with the cooperative approach, is that the neighbors will to need to have the means to reduce the amount of traffic they are sending to the overloaded servers. To achieve this they will either have to drop or reject the traffic themselves by using some standalone overload control scheme or by informing their neighbors to reduce the traffic that is destined to the overloaded server. Informing the neighbors to reduce the amount of sent traffic is theoretically more favorable as it would push the excess traffic even further from the overloaded server. However, SIP servers do not usually keep track of how much of their traffic is traversing some server. Assume server C in Fig. 21.4 is overloaded and that it informs server B to reduce the amount of traffic it is sending. Server B can reduce the amount of traffic sent to server C by either using a standalone overload control scheme or informing server A to reduce the traffic sent from A to C. However, this would mean that server B needs to keep track of which upstream neighbor is sending how much traffic to each downstream neighbor. While this already makes the logic of a SIP proxy more complex it is still not sufficient. Consider the scenario in Fig. 21.4. Server A is sending traffic to servers C and D through server B. When Server C gets overloaded it asks server B to reduce its transmission rate by X. Assume X was an absolute number, i.e., say 10 calls. Server B will in its turn ask server A to also reduce its transmission rate by X, i.e., 10 calls/s. Now, server

A was sending 20 calls/s to server B with half of the calls going to server C and the other half going to server D. If A reduced its transmission rate by 10 calls/s then the actual load reduction on server C will only be 5 calls/s. Now consider *X* was a percentage value, i.e., server C asks server B to reduce its transmission rate by 10%. When server B asks server A to reduce its transmission rate by 10%, the rate of calls arriving at server C will be reduced by the needed amount. However, server A will also reduce the amount of calls arriving at server D by 10%. Hence to actually achieve the needed reduction at server C, server A will need to keep track of which calls actually traverse server C. This is only readily available if servers A and C are end systems, i.e., voicemail servers, PSTN gateways, conferencing server or a B2BUA.

So in short, the cooperative model is only effective in pushing the overload one hop upstream. Pushing the traffic further upstream is only possible in certain scenarios and could make the SIP components rather complex otherwise.

Hence, to be on the safe side a SIP server should implement a combination of both standalone and cooperative approaches. The SIP system will inform its neighbors about its overload situation. Monitoring the neighbors and assigning a certain share to them is also necessary in order to maintain fairness between the neighbors. Simply asking all neighbors to reduce their load by a certain percentage for example might be the simplest kind of explicit notification information to use but will mean that all neighbors will reduce the amount of traffic they are sending to the overloaded system. This would be unfair towards neighbors who are not sending a lot of traffic to the overloaded systems and are hence not the cause of the overload.

21.4 Conclusions

Handling overload at SIP servers is an issue that is being discussed and researched both in academia and standardization bodies. This paper looks at the main ingredients of overload control algorithms and discusses the major pros and cons of the currently investigated schemes in the literature. With first discussions and publications already more than four years old the topic is no longer fresh. However, till now there has been no single solution that provides a proper answer to all the different facets and scenarios of overload control. It is actually questionable if one single approach can solve all related issues. Algorithms dealing with the communication between user agents and proxies will be different than those dealing with the communication between proxies. Overload control when UDP is used as the transport protocol will often look differently from the cases when TCP or SCTP is used. While standardization bodies can provide the proper mechanisms for enabling a cooperative kind of overload handling, the logic used at different SIP components will most likely be different depending on the type of the component, e.g., proxy versus application server, the deployment scenario and the vendor's preferences. Further, most of the approaches in the literature assume that overload was caused by non-malicious reasons, e.g., flash crowds. The design of

the overload control scheme should however include monitoring mechanisms that enable it to distinguish between malicious and non-malicious overload reasons. Without such distinction both malicious and non-malicious users will be punished in the same manner which will lead to unsatisfied subscribers.

References

1. Ohta, M.: Simulation study of SIP signaling in an overload condition. In: Hamza, M.H. (ed.) Communications, Internet, and Information Technology, pp. 321–326. IASTED/ACTA Press (2004)
2. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session initiation protocol. RFC 3261 (Proposed Standard). Updated by RFCs 3265, 3853, 4320, 4916. http://www.ietf.org/rfc/rfc3261.txt (2002)
3. Kuthan, J., Ehlert, S., Sisalem, D.: Denial of service attacks targeting a SIP VoIP infrastructure—attack scenarios and prevention mechanisms. IEEE Netw. Mag. **20**(5), 26–31 (2006)
4. Rosenberg, J.: Requirements for management of overload in the session initiation protocol. RFC 5390 (Proposed Standard) http://www.ietf.org/rfc/rfc5390.txt (2008)
5. Gurbani, V., Hilt, V., Schulzrinne, H.: Session initiation protocol (SIP) overload control. Internet Draft, Internet Engineering Task Force (2010) (work in progress)
6. Hilt, V., Noel, E., Shen, C., Abdelal, A.: Design considerations for session initiation protocol (SIP) overload control. Internet Draft, Internet Engineering Task Force (2010) (work in progress)
7. Shen, C., Schulzrinne, H., Nahum, E.: SIP server overload control: Design and evaluation. In: Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM08) Heidelberg, Germany (2008)
8. Signalling flows for the IP multimedia call control based on session initiation protocol (SIP) and session description protocol (SDP). Technical specification group core network and terminals, 3rd Generation Partnership Project (2007)
9. Charles Shen, Henning Schulzrinne: On TCP-based SIP Server Overload Control. In: Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM10) Munich, Germany (2010)
10. Sisalem, D., Floroiu, J.: Protecting VoIP services against DoS using overload control. In: NorSec 2008. Technical University of Denmark (2009)
11. Nahum, E., Tracey, J., Wright, C.: Evaluating SIP server performance. Research report RC24183, IBM T. J. Watson Research Center (2007)
12. Roach, A.B.: Session initiation protocol (SIP)-specific event notification. RFC 3265 (Proposed Standard). http://www.ietf.org/rfc/rfc3265.txt (2002)
13. Sisalem, D., Floroiu, J.: Voip overload, a senders burden. In: CCN10. Orlando, USA (2010)

Chapter 22
Towards Smarter Probes: In-Network Traffic Capturing and Processing

Nicola Bonelli, Andrea Di Pietro, Stefano Giordano, Gregorio Procissi and Fabio Vitucci

Abstract Monitoring is a matter of the greatest importance for the correct operation of current communication networks. In spite of that, analyzing and checking out the traffic flowing over a high capacity link is still a very challenging technological issue, due to the huge amount of data stemming from such a process. Furthermore, current national and international legislation is imposing stricter and stricter limits on the storage and utilization of potentially privacy-sensitive data that may be generated from monitoring applications. We argue that both of these problems can be effectively addressed by increasing and extending the capabilities of traffic capturing devices beyond plain packet capturing and flow metering. Therefore, we envision a new generation of smart probes that support traffic pre-processing according to the needs of the specific application that is expected to provide the final results of the monitoring activity. The benefits of such an approach are two-fold: on one hand, in-network traffic filtering allows to discard a huge amount of information which is not relevant at all to the selected application, thus relaxing the performance requirements of the application itself. On the other hand, traffic pre-processing can be used to hide personal information that may be

N. Bonelli · A. Di Pietro · S. Giordano · G. Procissi (✉)
CNIT and Dipartimento di Ingegneria dell'Informazione, Università di Pisa,
Via Caruso 16, 56122 Pisa, Italy
e-mail: Gregorio.Procissi@iet.unipi.it

N. Bonelli
e-mail: Nicola.Bonelli@iet.unipi.it

A. Di Pietro
e-mail: Andrea.Di Pietro@iet.unipi.it

S. Giordano
e-mail: Stefano.Giordano@iet.unipi.it

F. Vitucci
e-mail: fabio.vitucci@winmed.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_22,

made available only to a user in possession of the required privileges upon verification of a given condition. Following such a general approach we propose a modular architecture that allows application specific traffic pre-processing to be carried out in a scalable and performance-effective way. Such an architecture interacts with the external network by enforcing strict role-based policies, thus allowing selective and proportional information disclosure; the architecture as it is can be easily integrated with a standard access control infrastructure. An example application is demonstrated in order to prove the effectiveness of the proposal.

Keywords Network monitoring · Real time processing · Probe programming · Privacy-aware monitoring

22.1 Introduction

22.1.1 Why Smart Probes?

State-of-the-Art approaches do not scale. Many currently adopted monitoring applications (Snort [1] is just the simplest example) are built as a unique block, which takes as input a stream of raw packets (a trace, which can be made up of live traffic or traffic which has been previously captured by a probe) and returns the desired output. Usually, such systems leverage the standard PCAP interface, which provides a similar kind of access to stored and live data. Several examples of such solutions have been proposed in the literature. Coralreef [2] provides an API implementing two stacks to retrieve data from hetherogeneous sources: one of the stacks is used to import traces from different kinds of links while the second one enables working with flow records. The work [3], instead, proposes a large scale measurement infrastructure which is more tailored for active and performance measurements.

Nowadays, such a design paradigm shows several limitations. On one hand, with the current fast growth of link capacities and traffic volumes [4], having a full fledged monitoring application inspect every single packet on a multi-gigabit link raises huge performance issues. Common general purpose hardware hardly keeps the pace with the packet rates characterizing current core links, even when minimal per-packet processing is required. Hardware-based implementations, in turn, usually lack the flexibility and expressiveness which are required to implement complex traffic analysis applications. Moreover, according to the current technological trends, traffic speed is growing faster than processing power, so that having the application monitor a complete traffic stream on a core link will be increasingly problematic. Such a scaling problem surfaces, in particular, when dealing with distributed monitoring applications, which have to deal with data captured by multiple vantage points scattered across the network. This kind of applications is

likely to become more and more popular, as distributed anomalies and cyberthreats (of which botnets are a major example) require a detection/mitigation infrastructure which correlates events and alerts from several probes, possibly belonging to different domains.

Other monitoring applications address this issue by taking as an input pre–processed reports formatted according to NetFlow [5] or IPFIX [6] protocols. Such reports encompass summarized per-flow information (usually cumulative packets or bytes counters, duration and TCP flags) which are usuful for a number of applications (billing is a common example). IPFIX, in fact, offers a significant degree of flexibility in letting the user define the data types it needs to convey.

Such an approach is nowadays very popular in the field of distributed monitoring: a scenario where several NetFlow probes report to a centralized collector is common in many operational scenarios. However, despite achieving a significant reduction over packet traces, the practice of exporting a standard information for every flow as an input to the monitoring application still presents significant issues. On one hand, collecting per flow data on a full operator network is likely to raise a huge scalability problem, as the collector represents a serious performance bottleneck and is likely to get congested: the rate of new flows entering the network is likely to be one or two orders of magnitude lower than the packet rate but, in a large operator network, can easily rise to prohibitive figures. Indeed, much of the information which is conveyed by per-flow records is of little or no interest to the application, especially when it deals with detecting anomalies and security breaches: a famous Van Jacobson quote reports that "*... If we're keeping per-flow state, we have a scaling problem, and we'll be tracking millions of ants to track a few elephants*". In addition, per flow information is, in many cases, not detailed enough for the application's needs: as an example, Snort requires scanning the packet payload for malware signatures, while applications dealing with network path delays needs precise timestamps of certain packets.

In many cases, performance problems have been addressed by implementing some critical application primitives in hardware. While this solution is certainly effective in reaching the required throughputs, it significantly impacts on the flexibility of the monitoring devices: once the desired functionalities have been committed to the silicon, there is no way of updating them. DAG cards [7] are effective in capturing packets at high-speed, but usually only provide limited on-board filtering functionalities. Other probes export flow-data through NetFlow or IPFIX protocols; such information is sometimes not enough for certain applications. Some devices [8] use special purpose hardware in order to perform some specific monitoring tasks on high-speed links. However, as previously mentioned, they are intended for a specific task only and are not able to support a wide range of applications.

Privacy preservation is not an option. Another big issue that current network monitoring practice needs to address is the compliancy with current legislation in terms of privacy preservation. In particular, due to current legislation trends ([9] effectively explains the constraints imposed by EU legislation), much of the information which is retrieved from the captured traffic is considered privacy sensitive, and its disclosure and storage is subject to strict rules. Such constraints,

besides further preventing the practice of trace based monitoring, reflect on almost all of the above described approaches. Not only the packet traces, but a huge number of derived metadata (including per-flow reports) are considered to contain privacy sensitive information and therefore their export and storage is subject to very strict rules (if not completely forbidden).

A classical approach to privacy–aware network monitoring has been to run specific anonymization tools over the packet traces or the metadata to be exported and to hand them over to the legacy analyses, so as to have them work with "sanitized" input. A broad range of such techniques has been proposed in the literature (see [10] for a detailed survey), including blackmarking (complete removal of given fields), pseudonymization (substitution of a given identifier with an anonymized alias), timestamp approximation or prefix preserving anonymization (a technique allowing to replace IP addresses with a pseudonym which preserve their common prefix length relationships). Several tools have been published claiming to be able to sanitize a packet trace in a user configurable manner. However, several studies show that classic anonymization schemes may be easily reversed by skilled attackers using statistical analysis techniques (see, for example [11] or [12]). A theoretical study [13] showed that there is a clear trade-off between the information content of a trace and its ability to preserve privacy: as a consequence, traces which are well anonymized turn out to be almost useless for a monitoring application. In addition, anonymizing the data before handing them over to the monitoring application increases the burden of traffic capturing, so making the performance issues even more serious.

Finally, particular care has to be taken in distributed monitoring scenarios, especially when dealing with multi-domain applications: in the latter case, indeed, beside complying to the law, the application must ensure that no business confidential information is leaked to a possible competitor.

22.1.2 The Analyze While Capturing Paradigm

As opposed to traditional approaches in monitoring application design, in this work we propose to address both of the above illustrated major issues through in-network traffic processing performed by smart probes. In fact, traditional probe devices are usually designed with a strong focus on capturing performance but with little flexibility in terms of packet processing and exporting. On the contrary, we argue that the probe should be a flexible and programmable device that can filter and export the information it handles in a way that is specifically "matched" to a certain monitoring application. Of course, this means that a modular and extensible probe design is needed in order to accommodate different monitoring applications concurrently. An application specific filtering module directly on the probe can select and export the only information which is relevant to that particular application, thus enforcing immediate data reduction.

Such basic principle of "processing while capturing" is beneficial to the overall monitoring infrastructure in that it addresses at the same time the two main issues which have been described. In particular, it allows to support:

- *Performance scalability through data reduction*. Information which is of no interest to the application is discarded, thus aggressively reducing the amount of data to be processed.
- *Selective data protection*. Personal information is hidden and is not allowed to leave the probe, unless some particular condition is met that makes information disclosure necessary and therefore legal.

As for the latter point, our approach effectively leverages the so-called "proportionality principle", which is common in privacy related legislation: an application is allowed to receive only the data which is strictly necessary to its operation. On-probe filtering, therefore, allows the data which are required by a given application to be legally exported out: the privacy of the users is preserved as most of the regular and legitimate traffic (which is of no interest to the applications) will never leave the probe, thus preventing any possible leak of sensitive information.

A simple example of how these principles are implemented is that of a common intrusion detection application: in that case the filtering module on the probe will perform a fast scan of every packet, in order to separate legitimate traffic (the vast majority), from suspicious flows, which will be sent to the actual application for more detailed inspection. Of course, such a fast filtering activity requires proper algorithmic, as it will be illustrated in Sect. 22.5.

A smart probe, in addition, can export information in very compressed and anonymity-preserving data structures, which allow to detect a certain class of events without leaking information about the single users; *sketches* [14] and *Bloom filters* [15] are good candidates for this kind of solutions.

Such a general principle is embodied in a flexible monitoring probe architecture that will be described in the next sections. Such architecture allows to encompass several instances of application-specific processing running in parallel in a scalable and effective way. This is accomplished by means of several architectural choices:

- Decoupling of control functions (access control, configuration, etc.) from traffic processing functions.
- Distribution of the traffic processing workload among several (possibly heterogeneous) processing units.
- Dynamic allocation of the computational resources.
- Strict access-control mechanisms with role-purpose specifications.

Such a novel architecture has been adopted and deployed within the integrated prototype built as a result of the FP7 European research project PRISM [16] (under which this research activity has been carried out), where it was used as the base architecture of the Front-End module [17]. The goal of such a research was to design a framework allowing to deploy heterogeneous off-the-shelf monitoring applications while respecting user privacy constraints.

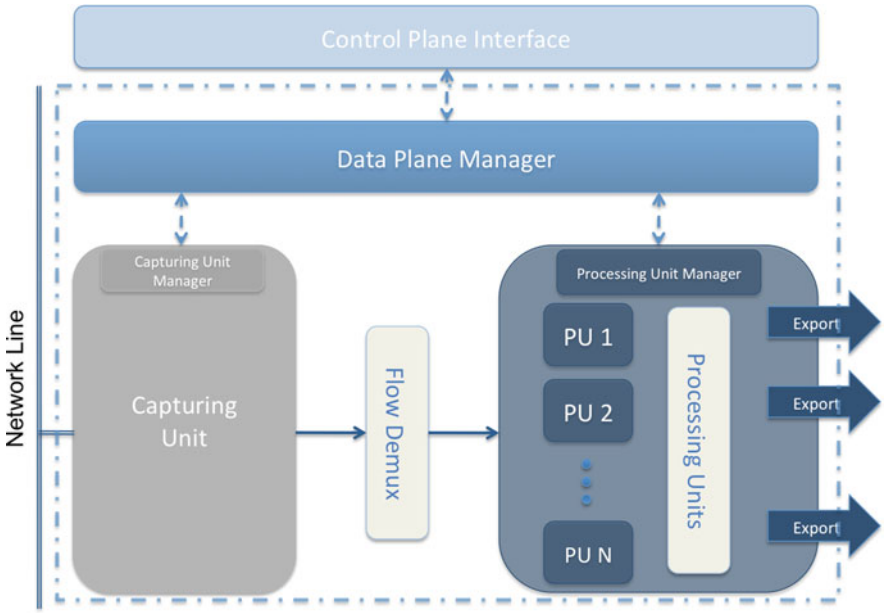


Fig. 22.1 Overall probe architecture

The rest of the chapter is organized as follows. Section 22.2 describes the overall probe architecture while Sects. 22.3 and 22.4 deal more specifically with the data and control planes, respectively. Section 22.5 is devoted to advanced algorithmic for on-the-probe traffic processing while Sect. 22.6 reports a practical use case deployment of the described probe architecture. Finally conclusions end-up the contribution.

22.2 The Probe Architecture at-a-Glance

The probe design which embodies the overall previous discussion is represented in Fig. 22.1. The architecture reflects the idea of an advanced logical component that is in charge of both capturing data at gigabit speed and performing a set of basic operations on-the-fly in an efficient way in order to:

- Isolate relevant flows out of the set of all traffic flows
- Extract relevant information from the observed traffic and eliminate irrelevant information (data reduction)
- Protect relevant information to force compliancy with end-users privacy requirements.

The device directly connected to the wire is the Capturing Unit (CU). Its main functions are: capturing traffic, timestamping packets, classifying packets and

sending snapshots of the captured packet to multiple destinations (Processing Units). As Fig. 22.1 highlights, multiple Processing Units (PU) are supported. They are deployed as commodity PCs or dedicated HW devices and they receive data from the CU and are in charge of actually implementing the application-matched data processing and protection (*analyses*). After their operations, PUs export results towards the further stages of the monitoring applications. The overall system configuration is controlled by the Control Plane Interface that issues commands to the Front-End Data Plane Manager (DPM), which, in turn, communicates with the other components. The DPM, in particular, enforces the received directives on the CU and the PUs by instantiating commands to the Processing Unit Managers and the Capturing Unit Manager, which constitute the control components of the traffic processing blocks.

22.3 Probe Data Plane

22.3.1 Capturing Unit

As already mentioned, this is the component which is in charge of capturing the monitored traffic and demultiplexing it among the analyses performing application-specific processing. In particular, packets can be dropped or selected for further processing, in which case a portion of them will be forwarded to one or multiple PUs, according to the rule table. As it is in charge of handling a potentially large data flow, it is likely to be implemented as a hardware accelerated device. Depending on the monitoring application requirements, this unit may extract header values by stripping only the necessary information from packets (in the simplest case this might be even just packet truncation after the layer the header), thus alleviating the workload on the upstream PUs. In order to deliver the captured traffic to the proper processing block, an internal interconnecting network is used. A simple and effective implementation consists of an Ethernet network where *batch frames* are transmitted, consisting of snapshots of captured traffic data plus some extra meta-data information.

22.3.2 Processing Units

As far as the data plane concerns, processing units are physical devices that receive data from the capturing unit through standard interfaces (standard libpcap interfaces), process them according to the analysis function(s) installed on them, and finally deliver encrypted data to the external world. PUs are usually software-based devices, where different kinds of processing are dynamically allocated. However, in case processing intensive analyses are required (e.g., deep packet inspection), they can use special purpose hardware. Such particular features are taken into account by the control plane when resource allocation has to be performed.

A typical PU has to implement several functionalities, which are described in the following.

Abstraction layer. In order to provide a standard interface between the proprietary protocol implemented by the CU exporting process and the analysis functions, a proper abstraction layer is installed on the PUs. Such an abstraction layer restores the compatibility between the proprietary batch frame format and the standard libpcap capturing interface. The abstraction is typically implemented by leveraging the concept of virtual capturing interface: all of the packets belonging to the same group (matching the same set of flow definitions) can be received by the analysis function through a virtual network interface, just as if the traffic were captured by the PU itself. The major advantage of this approach is that it totally hides the underlying batch framing process and allows compatibility to existing software. Therefore, brand new applications that reside on PUs can be designed and implemented in a completely independent manner, as they will just need to rely on libpcap.

Analysis Functions. Analysis functions are applications developed at the user space that (i) read (possibly truncated) packets from the virtual monitoring interfaces made available by the abstraction layer, (ii) Process data according to their specific function and (iii) export their outcome to the further processing stages of the monitoring application.

The development of analyses actually implements the overall design philosophy of further reducing data directly at the probe (the first application-agnostic stage of reduction occurs at the capturing unit level as packets are truncated to a customizable size) and delivering to the following stages the minimum necessary information only. Although the functional interface of such analyses is quite simple, they are subject to strict performance requirements, as they must process packets in real time and by keeping a very limited amount of state.

As it will be elaborated upon in Sect. 22.5 the use of probabilistic data structures, such as *Bloom filters*, that keep state in a compressed and quickly accessible way, is envisioned at this stage of the application.

Besides processing traffic, the PUs are also directly responsible for conveying it out of the probe to the further stages. The destinations of such reports (as well as some formatting options) are communicated to the analyses through the control plane. In principle, any analysis function may use its own protocol, even if the use of standard formatting (as the IPFIX protocol, that was the choice for the PRISM project) is recommended.

22.3.3 Data Plane Performance

The probe data plane is subject to very strict real-time constraints, as a large portion of the traffic flowing over the monitored link has to be conveyed through it (potentially, as some flows may need to be duplicated to several PUs, the traffic rate might be even higher than that on the link). In order to prove that such a component can be actually implemented and meet such performance requirements,

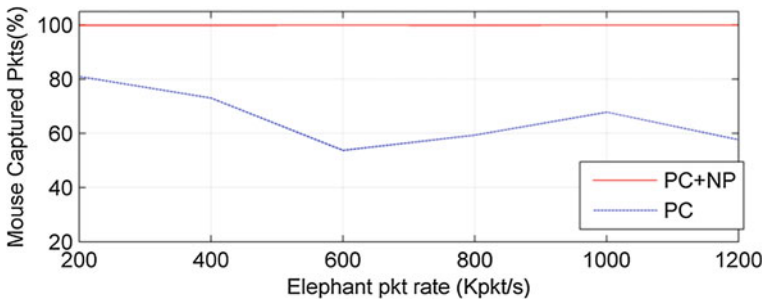


Fig. 22.2 Fraction of captured packets

we report here the experimental results as obtained for the PRISM implementation of the smart probe.

The capturing unit device is implemented over a Network Processor (NP) architecture, thus strongly leveraging the hardware parallelism provided by such devices. In particular, the fast data path functions of an NP are implemented by dedicated RISC processors, called microengines (MEs). In this case, our reference platform is an Intel IXP2400 NP, which is provided with 8 MEs that can be arranged to process packets both in a pipelined and a parallel scheme. Such MEs communicate by exchanging pointers to the packets to be processed, which are stored into an External DRAM memory block, thus implementing a zero-copy scheme. Our chip is hosted on a Radisys ENP-2611 evaluation board, equipped with three 1 Gb/s interfaces. In the experimental testbed, the NP-based capturing device was connected to a standard commodity PC which captures traffic through an off-the-shelf *tcpdump* application. Data streams are generated by Spirent ADTECH AX4000 hardware packet generator and analyzer.

Our experiment aims at showing the capabilities of the system in extracting and processing a *mouse* flow (generated at a rate of 50 kpkt/s) in presence of an *elephant* flow (generated at increasing packet rates). The compound flow is captured by the PC alone and through the NP.

Figure 22.2 shows the full advantage obtained by using the NP in the flow extraction. The entire mouse flow is captured by the NP-based system while it shares the fate of the elephant flow when captured by the PC alone. In this context, the PC shows all its architectural flaws in that it loses a huge amount of packets, while the NP-based system performs this operation without loss.

22.4 Probe Control Plane

The high level scheme of the probe control plane is depicted in Fig. 22.1. An analysis function is dynamically set up on the probe upon request from an external entity (be it the monitoring application itself or a further processing stage). All kind of requests are received through the Control Plane Interface (CPI), which acts

as the border module between the probe itself and the rest of the system. Typical requests being served at the CPI and properly mapped and delivered to the Front-End include *setting up*, *tearing down*, *stopping* and *restoring* an analysis, as well as *updating* its parameters (output format, report destination, etc.). Naturally, strict authentication and authorization mechanisms have to be enforced within this block, ranging from standard X.509 identity and privilege management infrastructure to more involved schemes, such as the *purpose-role* based mechanisms used within the PRISM project [18].

Once a request has been authorized, it is taken over by the Data Plane Manager (DPM). The DPM is the central component in charge of managing the capturing unit and the available processing units. Its main purpose is to arrange and launch the analyses on the various PUs (by communicating to the Processing Unit Manager installed on all PUs), and to configure the CU (through the Capturing Unit Manager) to capture and forward them the portions of traffic that will be the subject of the analyses. Since DPM has a perfect knowledge of the status of all the processing units, before setting up new analyses it first checks for resource availability (*admission control*) and may enforce load balancing in order to optimize resource usage.

Any new request is accompanied by a tuple (source and destinations networks, ports, protocol) that specifies the traffic flow subject of the inspection. Such information is used by DPM to dynamically configure the CU, in order to classify and forward the traffic to the selected PU. It is worth noticing that, since the traffic classification rules may overlap (i.e.: two distinct analyses running on top of the same PU may have a network address range in common), set theory results applied to collections of ranges have been efficiently used to minimize the amount of traffic to be forwarded from the CU to the various PUs. Indeed, DPM relies on a generic multidimensional range algorithm to expand and reduce the classification rules in term of generic tuples, as well as to arrange them to fit with the longest prefix match algorithm implemented in the CU.

22.5 On-the-Probe Advanced Processing Techniques

The development of a probe that plays an active role in the overall monitoring process increases the burden on such component, and therefore requires the adoption of a novel design paradigm where methodologies and functionalities are strongly aware of the available hardware. Indeed, in order to accomplish operations like capturing, classification, anomaly detection, flow discrimination and isolation, etc., *at wire speed*, a detailed knowledge of the hardware capabilities/bottlenecks, as well as the fine grained analysis of the available time budget for each micro-operation involved are required.

The attempt to come up with performance effective solutions to be integrated into the front-end stage must then pursue the investigation of *stateless* and *memory*

saving approaches with *constant look-up time* in that they tightly reflect into faster operations since they can take advantage of layered caches available in today's off-the-shelf multicore processors.

Given this, a very promising approach towards packet processing and inspection is based on *Bloom filters* (BFs) [15] and their variations. BFs are compact and fast data structures for approximated set-membership query and their popularity is rapidly increasing because of their very limited memory requirements, trading certainty for predictability and efficiency in time and space. A BF represents a set of n elements by using a bitmap of m elements. Each element of the set is mapped to k elements of the bitmap whose position is given by the result of$k$ hash functions. To check whether an element belongs to the set, one just needs to evaluate the k hash functions and verify if the corresponding bits of the bitmap are all set. As the hash functions of different elements may collide, the filter may allow for false positives.

Counting bloom filters (CBFs) are a simple extension that implement counters by having more than one bit per bin. They therefore support both insertion and deletion of elements by counter incrementing and decrementing, respectively. The use of CBFs for statistical data processing turns out to be extremely flexible although the fixed size of bins may cause memory inefficiency. A significant improvement can be obtained by allowing dynamic size of bins, compression, and multi-layering. These modifications, as described in [19] and [20], appear applicable to the data processing performed by the probe.

For example, let us consider a set of rules used to classify flows at the front-end: a CBF can easily be used to represent the set. In order to verify whether a packet obeys one of the rules of the set, a simple lookup operation consists of evaluating k hash functions and comparing the values in all resulting bins to zero. If the result is positive, the packet satisfies the rule with small and predictable error probability, and can be exported for further processing.

Several papers have been published that describe in detail the application of such technique to specific monitoring applications. Among them, [21] and [22] have been devised within the PRISM research project.

22.6 Actual Monitoring Applications: A Practical Use Case

In order to better illustrate the above discussed features, we describe here a possible use case scenario where the proposed architecture allows to meet high-performance demands while being privacy preserving at the same time.

Let us assume the smart probe is used to protect a company network from external attacks by monitoring its gigabit ingress/egress link. In particular, let us suppose that two monitoring applications are used: a *scan detection* application that flags anomalous behaviors, and a TCP SYN *flooding detection* application. Both applications would raise significant performance issues if they had to process the whole traffic flowing through the link. In order to avoid that, special pre-filtering

functions are installed on the Capturing Unit: the scan detection application will receive only the headers of the traffic entering the network, while the TCP SYN flooding detector will be fed with the headers of both incoming and outgoing TCP segments. The classified traffic is shipped by the CU onto batch frames and forwarded to separate PUs; at this stage, the data rate turns out to be significantly reduced. With a standard trimodal packet length distribution, a quick back-of-the envelope calculation shows that each processing unit needs to process less than 20 MBps of traffic, which is affordable with current off-the-shelf hardware.

The amount of flows to be processed, however, has not been reduced and keeping per-flow state is still unfeasible, due to its excessive memory footprint. To this end, PU processing should be carried out in a quick and stateless manner, which can be achieved by using probabilistic data structures. In particular, the method proposed in [23] provides a good heuristic for TCP SYN flooding detection while the one proposed in [22] can be adopted for fast stateless scan detection. Such a second-stage filtering operated at the PUs, is used to select suspicious traffic, which can be legitimately conveyed to an external collector for further analysis (as the volume of such data is likely to be very low, stateful and more complex analyses can be carried out), and discard legitimate traffic, which is likely to contain privacy sensitive information and will never leave the probe.

22.7 Conclusions

In this work we introduced the idea of a smart probe, which, besides standard traffic capturing functionalities, provides support for application-matched traffic pre-processing. We argued that such an approach allows both to increase the application performance (by having it process a selected subset of the captured traffic only) and to perform monitoring in a privacy preserving way. After illustrating the underlying design philosophy, we described a modular and extensible architecture that embodies such general concepts in a resource effective way. Such a design is based on a two-stages traffic processing, which are managed by a proper control plane. The capability of such an architecture to support heterogeneous monitoring applications in a privacy-aware way while not degrading their performance and functionalities has been assessed within the PRISM research project, where such architecture has been used to build the traffic capturing block.

References

1. Snort. http://www.snort.org
2. Keys, K., Moore, D., Koga, R., Lagache, E., Tesch, M., Claffy, K.: The architecture of coralreef: an internet traffic monitoring software suite. In: In PAM (2001)
3. Paxson, V., Mahdavi, J., Adams, A., Mathis, M.: An architecture for large scale internet measurement. IEEE Commun. Mag. **36**(8), 48–54 (1998)

4. Cisco: Approaching the zettabyte era. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481374_ns827_Networking_Solutions_White_Paper.html (2008)
5. Cisco: Cisco systems netflow services export version 9. http://www.ietf.org/rfc/rfc3954.txt (2004)
6. Ip flow information export (ipfix). http://www.datatracker.ietf.org/wg/ipfix/charter/
7. Endace. http://www.endace.com
8. Palo alto networks. http://www.paloaltonetworks.com
9. Bianchi, G., Boschi, E., Gaudino, F., Koutsoloukas, E.A., Lioudakis, G.V., Rao, S., Ricciato, F., Schmoll, C., Strohmeier, F.: Privacy-preserving network monitoring: Challenges and solutions. In: 17th ICT Mobile and Wireless Communications Summit 2008 (2008)
10. FP7-PRISM: Deliverable d3.1.1 : State of the art on data protection algorithms for monitoring systems. Technical report http://fp7-prism.eu/images/upload/Deliverables/fp7-prism-wp3.1-d3.1.1-final.pdf (2008)
11. Hintz, A.: Fingerprinting websites using traffic analysis. In: Workshop on Privacy Enhancing Technologies (2002)
12. Bissias, G., Liberatore, M., Jensen, D., Levine, B.: Privacy vulnerabilities in encrypted http streams. In: Danezis, G., Martin, D. (eds.) Privacy Enhancing Technologies, Lecture Notes in Computer Science, vol. 3856, pp. 1–11. Springer, Berlin (2006)
13. Yurcik, W., Woolam, C., Hellings, G., Khan, L., Thuraisingham, B.: Privacy/analysis tradeoffs in sharing anonymized packet traces: Single-field case. In: ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security. pp. 237–244. IEEE Computer Society, Washington, DC, USA (2008)
14. Cormode, G., Muthukrishnan, S.: An improved data stream summary: The count-min sketch and its applications. J. Algorithms. **55**, 29–38 (2004)
15. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM. **13**, 422–426 (1970)
16. Fp7 prism. http://fp7-prism.eu/
17. FP7-PRISM: Deliverable d2.2.2: Detailed system architecture specification. Technical report http://telscom.ch/wp-content/uploads/Prism/FP7-PRISM-WP2.2-D2.2.2.pdf (2010)
18. Lioudakis, G.V., Gogoulos, F., Antonakopoulou, A., Mousas, A.S., Venieris, I.S., Kaklamani, D.I.: An access control approach for privacy-preserving passive network monitoring. In: Proceedings of the International Conference for Internet Technology and Secured Transactions, ICITST 2009. pp. 1–8 (2009)
19. Ficara, D., Giordano, S., Procissi, G., Vitucci, F.: Multilayer compressed counting bloom filters. In: Proceedings of the 27th IEEE Conference on Computer Communications, INFOCOM 2008. pp. 311–315 (2008)
20. Ficara, D., Giordano, S., Procissi, G., Vitucci, F.: Blooming trees: space-efficient structures for data representation. In: Proceedings of the IEEE Internaional Conference on Communications, ICC '08. pp. 5828–5832 (2008)
21. Antichi, G., Ficara, D., Giordano, S., Procissi, G., Vitucci, F.: Counting bloom filters for pattern matching and anti- evasion at the wire speed. Netw. Mag. Global Internetwkg. **23**(1), 30–35 (2009)
22. Bianchi, G., Boschi, E., Teofili, S., Trammell, B.: Measurement data reduction through variation rate metering. In: Proceedings of the IEEE INFOCOM. pp. 1–9 (2010)
23. Sun, C., Hu, C., Tang, Y., Liu, B.: More accurate and fast syn flood detection. In: Proceedings of the 18th Internatonal Conference on Computer Communications and Networks, ICCCN 2009. pp. 1–6 (2009)

Chapter 23
IDS Rules Adaptation for Packets Pre-filtering in Gbps Line Rates

Simone Teofili, Enrico Nobile, Salvatore Pontarelli and Giuseppe Bianchi

Abstract The enormous growth of network traffic, in conjunction with the need to monitor even larger and more capillary network deployments, poses a significant scalability challenge to the network monitoring process. We believe that a promising way to address this challenge consists in rethinking monitoring tasks as partially performed *inside* the network itself. Indeed, in-network monitoring devices, such as traffic capturing probes, may be instructed to perform intelligent processing and filtering mechanisms, so that the amount of data ultimately delivered to central monitoring entities can be significantly reduced to that strictly necessary for a more careful and fine-grained data inspection. In such a direction, this chapter focuses on the design and implementation of an hardware-based front-end pre-filter for the topmost known Snort Intrusion Detection System (IDS). Motivated by the practical impossibility to pack a large amount of legacy Snort rules over a resource-constrained hardware device, we specifically address the question on how Snort rules should be adapted and simplified so that they can be supported over a commercial, low-end, Field Programmable Gate Array (FPGA) board, meanwhile providing good filtering performance. Focusing on about one thousand Snort rules randomly drawn from the complete rule set, we experimentally determine how these rules can be simplified meanwhile retaining a comparable detection performance with respect to the original, non adapted, rules,

S. Teofili (✉) · E. Nobile · S. Pontarelli · G. Bianchi
Consorzio Nazionale InterUniversitario per le Telecomunicazioni (CNIT)/University of Rome "Tor Vergata", Via del Politecnico 1, 00133, Rome, Italy
e-mail: Simone.Teofil@uniroma2.it

E. Nobile
e-mail: Enrico.Nobile@uniroma2.it

S. Pontarelli
e-mail: Salvatore.Pontarelli@uniroma2.it

G. Bianchi
e-mail: Giuseppe.Bianchi@uniroma2.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_23,

when applied over a “training” dataset composed of a relatively large traffic trace collected from a regional ISP backbone link. We then validate the performance of the adapted rules against additional collected traffic traces. We show that about 1000 adapted Snort rules can be supported over a low-end FPGA based Snort pre-filter, with 93% data reduction efficiency.

Keywords Online traffic analysis · SNORT · IDS · FPGA

23.1 Introduction

The constant growth in diffusion and performance of networks is accompanied to an increase in the number of hacking and intrusion incidents. Consequently Intrusion Detection Systems (IDS) has been proposed to detect the presence of such type of incidents. Most IDSs monitor traffic flows and inspect packet payloads for detecting predetermined attack patterns (signatures). Different kinds of intrusions may be taken into account: shell-codes that exploit Operating Systems vulnerabilities to gain unauthorized access to a host computer, policy violations in the use of a corporate network, port scans, and so on. Moreover, in recent years, IDS rules have been extended to detect also user mis-behavior, such as exchange of pornography material and so on. Obviously, the collection of IDS rules must be promptly updated in order to cope with emerging threats or monitoring needs, and this yields a number of rules that is constantly increasing both in cardinality as well as in rule complexity.

Open source, widespread deployed, Network Intrusion Detection System (NIDS) such as Snort [1] are software based. Because of software limitations, as well as limited traffic capturing capabilities of ordinary PC network cards, they are mostly used in relatively low loaded small networks, whereas their exploitation for backbone links is questionable, due to the huge traffic volume involved and the multi gbps line rates.

Hardware based NIDS have been proposed to face these limitations, and sustain traffic inspection at wire speed [2]. These systems are installed over traffic probes, and act as pre-filter. Their goal is to detect the subset of possible malicious streams, and reduce the amount of traffic data delivered to a back-end software NIDS. The effectiveness of such systems can be measured in terms of data reduction capabilities: a significant data reduction would bring about the possibility to reuse legacy software-based IDS and permit cheap deployments.

One on the best candidate technologies for the development of such hardware NIDS systems are the FPGA. These reprogrammable components are designed to be configured by the customer or designer after manufacturing, therefore they are called “field-programmable”. This devices can be programmed to accomplished the pattern matching activities needed by the NIDS at very high speed. Moreover, due to their reprogrammability, the set of rules that are checked can be easily updated downloading a new bitstream in the FPGA.

Designing an hardware-accelerated IDS pre-filter over these devices, is however a non trivial task. Indeed, IDS rules such as the Snort rules set, are not limited to "basic" string matching. Rather, they may include multiple types of matching (i.e., content matching, byte matching, uri matching), as well as they may require the matching of multiple contents further regulated by "modifiers", indicating the position in the flow in which the content must be located, or the distance between a content and the previous. Finally, some rules may require the matching of regular expressions. Even if several works [3, 4] have described a thorough FPGA implementation of complex rules, the amount of logic resources and the design effort needed to implement these rules on the FPGA can be overwhelming, especially when the goal is to move away from proof-of-concept implementations supporting a few tens of rules, and reach the practical real-world target of supporting an order of a thousand rules or more.

A practical solution to this issue consists in devising an IDS pre-filter which supports a set of *loosened* rules. The basic idea is very simple, and can be easily understood over the following trivial example. Assume that a Snort rule R_{orig} is triggered when two content patterns C_1 and C_2, for instance separated by a modifier, are matched. Consider now a new rule, R_{adapt}, devised to match only C_1. Its implementation would obviously require a lower amount of hardware resources. Moreover, its detection capability would be a *superset* of that of the original rule, and would not incur in *false negatives*, i.e., all cases detected by the original rule would also be detected by the new, "adapted", rule. The disadvantage of such rule adaptation is that the filtering performance clearly decreases, as more streams will be detected and delivered to the monitoring back-end for further inspections (false positives).

A trade-off emerges between filtering performance and ability to "pack" rules in the hardware front-end pre-filter. According to our in-field experience acquired during the experimental assessment work described in Sect. 23.6, there is no "obvious" adaptation mechanism which appears to optimize such a trade-off. For instance, if Snort rules were adapted so that only one single content were matched over the pre-filter, hardware implementation would be very efficient and simple, but the false positives rate may become unacceptably large. And to make things worse, filtering performance largely varies, and strongly depends on *individual* Snort rules.

To face these issue, we have resorted on an experimentally-driven, heuristic, rule adaptation methodology, made possible by our availability to access real traffic data delivered over the backbone of a regional ISP. Specifically, we have collected a number of large traffic traces, and used one of them as "training" set. We have then iteratively compared the number of alerts detected by a legacy Snort software, implementing a set of *original* Snort rules, with that detected by a pre-filter implementing *adapted* rules. At each iteration, we have identified which individual adapted Snort rules were the main cause of false positives (and why), and modified these rules accordingly. Performance assessment was then carried out by testing the adapted rule set over different network traces (up to 68 GB traffic).

The rest of the chapter is organized as follows. Section 23.2 provides the necessary background on Snort rules. Section 23.3 discusses the related work on

Table 23.1 Description of keywords modifiers

Modifier	Description
Offset: *N*	The search for the content begin after *N* characters
Depth: *N*	The search for the content ends after *N* characters
Distance: *N*	The distance between two contents is at least *N* characters
Within: *N*	The distance between two contents is less than *N* characters

hardware IDS filters. Section 23.4 presents our FPGA hardware implementation of the Snort pre-filter front-end. Section 23.5 details the Snort rules adaptation methodology, taking also into account how to adapt the rules so as the resulting hardware implementation is simplified. Section 23.6 presents experimental results performed over real traffic traces. Finally, conclusions are drawn in Sect. 23.7.

23.2 Description of Snort Rules

The Snort IDS performs deep packet inspection on incoming flows, checking whether some specific rules are matched, in which case an action is performed (i.e., alert, log or pass). Rules are divided into two logical sections: *rule header*, including packet header field information (source/destination IP, protocol, source/destination port), and *rule options*, containing alert messages, information on the patterns to match including their position inside the flow, etc. Moreover some keywords related to the state of the session and the direction of the packets (i.e., from or to the server) can be present. Almost all the Snort rules include one or more among the following keywords: content, modifiers, uri-content, and PCREs.

Content specifies a fixed pattern to be searched in the packets payload of a flow. If the pattern is contained anywhere within the packets payload, the test is successful and the other rule options tests are performed. A content keyword pattern may be composed of a mix of text and binary data. Most rules have multiple contents.

Modifiers identify a location in which a content is searched inside the payload. This location can be absolute (defined with respect to the start of the flow) or relative to the matching of a previous content. Snort modifiers are listed and described in Table 23.1.

Uricontent searches the normalized request URI field in the packet.

PCRE (*Perl Compatible Regular Expression*) define regular expression pattern matching using the PERL syntax and semantics. Even if PCREs give high flexibility to the description of the content to be searched for, the hardware implementation of a generic PCRE may require a lot of logic resources [3, 5]. In the Snort rules set, PCRE are usually used to describe some particular kind of attack. For instance, the following rule describes a buffer overflow:

```
alert tcp EXTERNAL_NET any- > HOME_NET 1655 (msg:''EXPLOIT
ebola USER overflow attempt''; flow:to_server, established;
content:''USER''; nocase; pcre:''/^USER\s [^\n]{49}/smi'';)
```

The PCRE search for a string starting with "USER", followed by a space (the escape sequence \s), followed by a maximum of 49 characters that differ from the newline character. If the rule is matched the IDS detects a too long user name that corresponds to a tentative buffer overflow.

The flexibility of the Snort rules corresponds to a difficulty in its hardware implementation. As will be discussed in the next section, the implementation of modifiers, of PCRE and of the other keywords could require a big effort and a huge amount of hardware resources.

23.3 Review of FPGA Based IDS Systems

In this section a brief overview of FPGA based IDS systems is presented. The aim of the section is to identify some common issues of these hardware implementation that we propose to solve by our rule adaptation procedure.

Packet pre-filtering has been originally proposed in [2]. This work exploits Bloom filters implemented on FPGA to detect malicious contents inside the inspected packets. The main limit of this approach is related of the inflexibility of this string matching approach.

More flexible implementations of pattern matching can be obtained by using Deterministic Finite Automata (DFA) [6] or Nondeterministic Finite Automata (NFA) [7]. These solutions can be applied to pattern matching and extended to matching regular expression such as PCRE [3, 4, 7]. Implementations based on DFA or NFA rapidly grow in size when the number of rules to be checked increase, and when Snort modifiers or PCRE are considered. Indeed, a DFA for a regular expression of length L can have $O(2^L)$ states [8]. NFA avoids state explosion but requires a more resource consuming hardware for its realization [7]. As such, DFA or NFA based pre-filter implementations challenging to support thousands of Snort rules may be very costly.

Our developed architecture is based on an alternative approach presented in [9], called shift-and-compare. It can give better results in the FPGA implementation, compared to DFA/NFA ones, since the basic logic block of an FPGA can be configured as a shift-register, reducing the amount of resources needed to implement this architecture. In particular, the shift-and-compare architecture allows sharing a significant part of memory elements between all the rules, thus enabling the implementation of thousand of rules on the same FPGA. Moreover, by using suitable registers, as we will show in details in Sect. 23.4, is possible to extend the shift-and-compare architecture for complex rule matching.

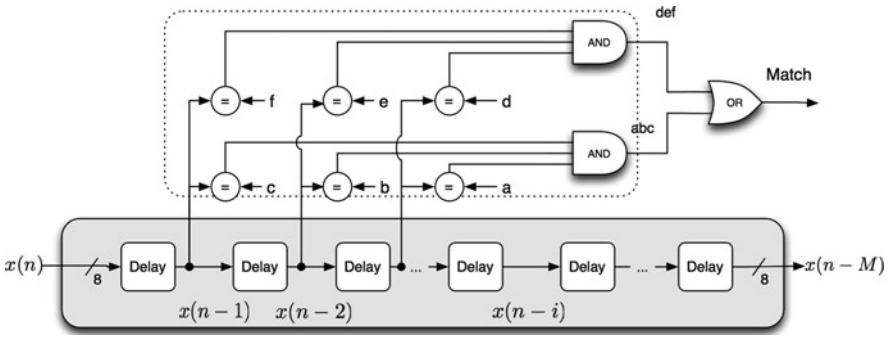


Fig. 23.1 Basic implementation of a multi-string matching block

Summarizing, the problem related to the current available FPGA based IDS systems are:

- Content matching alone is insufficient: in several cases, rule modifiers must be already accounted for by the FPGA pre-filter, in order to avoid a large amount of false positives
- The amount of logic resources grows linearly with the number of contents to be matched
- The most consuming logic resources are the memory elements representing the state of partially matched complex rules.

23.4 Snort Pre-filter Hardware Architecture

Our Snort pre-filter has been implemented over a Xilinx FPGA Virtex II Pro. FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow to connect together the logic blocks to perform more complex functions. Logic blocks are implemented by programmable Look-up Tables (LUT), that can be configured to perform combinational functions, like AND and XOR and also include memory elements, which may be simple flip-flops or more complete blocks of memory. Both the configuration of the logic blocks and of the interconnection are stored in a binary file called bitstream. The Virtex II pro FPGA has been integrated in a low-end PCI card equipped with an FPGA, 4 Gigabit Ethernet ports, SRAM and DRAM banks, called NetFPGA [10]. Even if the FPGA used by this board is obsolete, it is a suitable candidate for fast prototyping.

The content matching hardware illustrated in Fig. 23.1 follows the approach presented in [11]. Input bytes enter in a flip-flop chain. The longest content to be matched sets the maximum length of the flip-flop chain. The last M entered bytes are stored in the chain and can be evaluated in parallel. The evaluation is

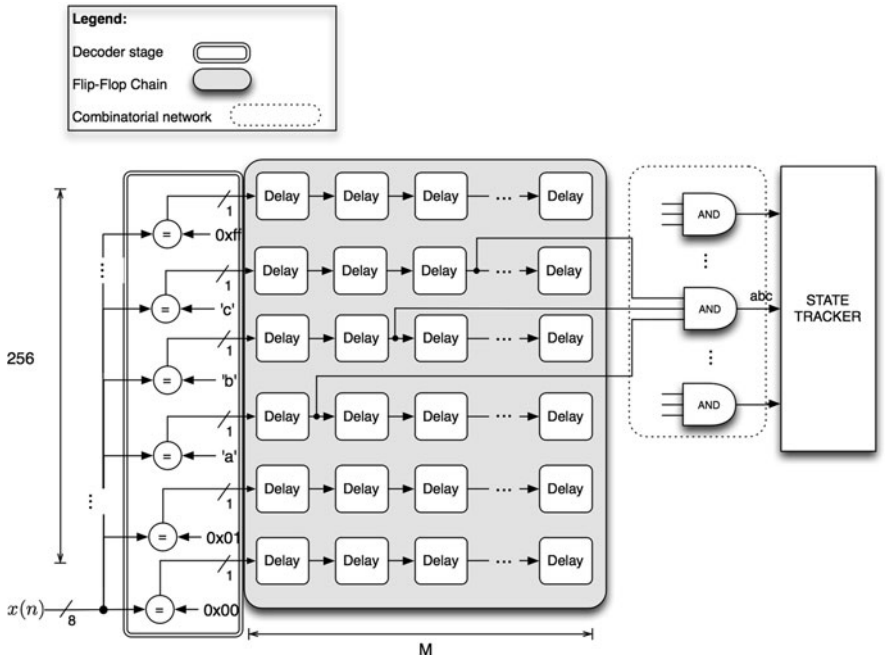


Fig. 23.2 Implementation of a multi-string matching block with decoded input

performed by the combinatorial circuit (shown in the dashed box of Fig. 23.1). For each content, the combinatorial network checks if each single character corresponds to the expected one and performs the logical AND of all the found characters. For instance, suppose that we are interested in matching the content *def*. We check if the third character is equal to '*d*', and the second is equal to '*e*' and the first is equal to '*f*'. If all these matches occur, a signal called *match* is raised.

When the number of rules increases, also the number of comparators, and therefore the resource occupation of the combinatorial circuit, increases. This extra complexity negatively affects the maximum operating frequency of the multi-string matching block. To overcome such a limitation, the bytes' comparators can be shared between the different string to be matched, as presented in [12], where the implementation of a multi-string matching block with decoded input delay chain is presented. The circuit implementing the decoded multi-string matching block is reported in Fig. 23.2.

The number of flip-flop is greatly increased, while the number of logic resources is decreased. But, as we already noticed before, the FPGA is able to optimal pack the chain of shift-registers in logic blocks (a LUT can be configures as a chain of 16 flip-flops), achieving only a limited resource occupation when long chains of shift registers are used. When the number of strings to be matched increases, this decoding circuit gives better results both in terms of area occupation

Table 23.2 Synthesis results for the different implementations of multi-string matching circuits

		200 rules	400 rules	800 rules
Basic (Fig. 23.1)	# of flip flops	508	1063	1302
	# of LUTs	1676	4301	6506
	# of slices (utilization [%])	908 (3%)	2315 (9%)	3459(14%)
With decoder stage (Fig. 23.2)	# of flip flops	1749	4371	4726
	# of LUTs	783	1780	3419
	# of slices (utilization [%])	769 (3%)	1847 (7%)	2618 (11%)

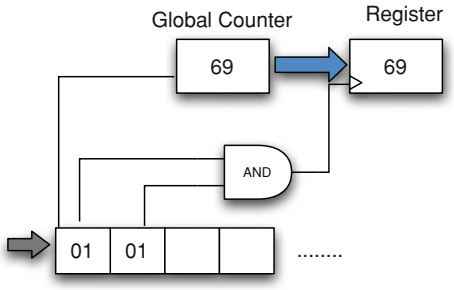


Fig. 23.3 First content matching

and achieved frequencies. The results in terms of resource occupation (LUT and FF) of such two implementations are reported in Table 23.2.

Our rule matching engine has been implemented by using the Xilinx Virtex II Pro XC2V50 FPGA available on the NetFPGA [10]. The multi-string matching circuits has been synthesized for three sets of rules, corresponding to 200, 400 and 800 rules extracted from the Snort ruleset. The results presented confirm that the basic architecture, without decoded stage, requires the highest number of LUTs and the lowest number of Flip-Flop. Instead, the sharing of the decoding operation allows savings around 50% of used LUT, but with an high cost in terms of Flip-Flops.

We have extended this basic content matching architecture with the capability to support more complex rule matching including Snort modifiers and simple PCREs. A global counter and dedicated registers were added to track partial matches. Figures 23.3 and 23.4 show an example where the rule to be matched includes a modifier "within":

```
content:''|01 01|''; content=''|00 00 02|''; within 10;
```

In this example, two binary contents |01 01| and |00 00 02| must be at a distance less than 10 bytes. The first part of this rule, i.e. the match of the content |01 01|, is performed by the two inputs AND gate. When the content is matched the value of the global counter is stored in a register (i.e. Fig. 23.3). Now, when the second content is matched, the system also checks if the difference between the global counter and the value stored in the register is less than ten bytes (i.e., Fig. 23.4). This extension is resource consuming because a register and a comparator must be instantiated for each part in which the rule is decomposed.

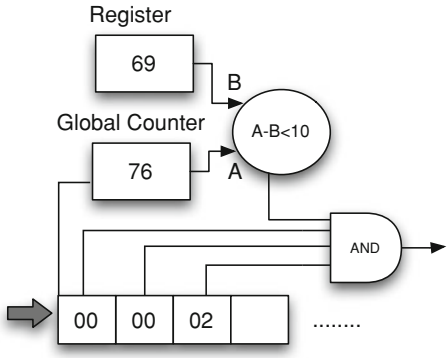


Fig. 23.4 Second content matching and modifier check

Table 23.3 Resource occupation of the string matching engine

Feature	
RAMB	0 out of 232 (0%)
Slices	6684 out of 23616 (28%)
Maximum frequency	125 MHz

Table 23.3 reports the hardware resource occupation after the synthesis of one thousand rules adapted as described in Sect. 23.5 Only 28% of the total FPGA logic resources were used. Note that the rest of the FPGA logic in the proposed framework is used to implement the modules necessary to manage the packets transmission or reception, instantiate Ethernet interfaces and debug operations as reported in [10].

23.5 Rule Adaptation

A major goal of our work consisted in the identification of how to adapt Snort rules for obtaining an efficient pre-filter hardware implementation meanwhile retaining a limited amount of false positives. In details, a crucial aspect is the identification of which "part" of the rules is most effective in detecting the specific attack. At least in principle, this analysis should be carried out at the level of *individual* rules, as, to the best of our knowledge, no general adaptation guidelines appear applicable, and indeed (as it will become clear in the rest of this section) a same approach applied to different rules provided widely different false positive performance.

We operated through the analysis of the set of Snort rules available in the Snort public distribution (5709 rules). We iteratively operated as follows.

First, we extracted from every rule the longest one among the content or uri-content included in each rule (in most generality, a rule specifies more than one matching). Note that this first step is somewhat analogous to what done in literature works in which only one content for rule is employed [2, 13]. The use of only one content per rule would permit an extremely efficient hardware implementation,

as it would allow to get rid of all the logic devised to cope with multiple contents, hence including registers and the comparators needed to track which contents have been previously matched for each rule. As an example, the following legacy Snort rule:

```
alert tcp, HOME_NET any -> $EXTERNAL_NET HTTP_PORTS
(msg:''SPYWARE-PUT Hijacker coolwebsearch.cameup runtime
detection''; flow: to_server, established; uricontent:
''svc=''; nocase; uricontent:''lang=''; nocase; uricontent:
''type=''; nocase; uricontent: ''mode=''; nocase; uricon-
tent:''art=''; nocase; uricontent:''acct=''; nocase; uri-
content: ''url=''; nocase;uricontent:''category=''; nocase;
uricontent: ''view=''; nocase; sid:6242; rev:2;)
```

was relaxed exploiting a single content[1]:

```
content:''category=''; nocase;
```

The list of so-relaxed rules (containing only the longest content/uri-content) was then tested against a "training" 7 GB trace containing a 40 min real network traffic captured over the backbone of a regional ISP. The resulting several thousand alerts signaled by Snort were then collected and re-attributed to each relaxed rule. In all cases, alerts were generated because the selection of the longest content resulted into a very common (or very short) string. For example, a content like "User-Agent:" is the longest one in many rules, but at the same time is a content present in all HTTP requests. The results of this automated analysis provided a set of long common contents[2] that cannot be used to represent a Snort rule. These contents were inserted into a so-called "blacklist". The described longest content extraction process was iteratively repeated, excluding blacklisted contents, (i.e., for a rule containing one of such contents the second longest one was chosen at the second iteration, and so on. The next step consisted in identifying all the rules that, after the content iteration described above, either included only a small content (2 or 3 bytes), or did not include any remaining content at all. For these rules, we selected via trials and errors (tested over the captured data trace) the modifier or simple PRCE starting from the simpler ones (from the point of view of an hardware implementation) available in the considered rule. For instance, in rule:

```
alert tcp $SMTP_SERVERS 465 -> $EXTERNAL_NET any (msg:''
SMTP SSLv2 Server_Hello request''; flow:from_server,
established; content:''|04|''; depth:1; offset:2; con-
tent:''|00 02|''; depth:2; offset:5; sid:3497; rev:4;)
```

the content was clearly not sufficient for an efficient matching. Indeed, the longest content (00 02) resulted in practical matches for almost all flows on port 465. The rule was therefore relaxed by exploiting the modifiers "depth" and "offset":

[1] To simplify implementation, the uricontent keyword was relaxed into a content keyword.

[2] The specific list being: {User-Agent, Server, Agent, Internet, Connection, complete/search?, /index.php.}

Table 23.4 Result for rule adaptation

	Number of packets	Number of flow
Original rule set	1534	511
Longest content only	86979	9745
Proposed adaptation	2831	857

```
content:''|00 02|''; depth:2; offset:5;}
```

Similarly, another example is rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 110 (msg:''POP3
USER overflow attempt''; flow:to_server,established;
content:''USER''; nocase; isdataat:50, relative; pcre:''/
^USER\s [^\n]{50,}/smi''; sid: 1866; rev:12;)
```

where the content "USER" alone cause a huge number of false positives; its relaxed version exploits the PCRE, i.e.:

IDS Rules adaptation for packets pre-filtering in gbps line rates

```
pcre'':/^USER\s [^\n]{50,}/smi''}
```

Unfortunately, this final step necessary to identify which part of the Snort rule is more effective, requires a work that only partially can be automated. The rules with the most representative part causing an excessive number of false positive have to be manually analyzed to understand how they can be best approximated.

The results obtained by our Snort rules adaptation process are shown in Table 23.4. The first row reports the packets and TCP flows identified as malicious by using the original snort ruleset. The second row presents the result by applying only the longest content of the rule. It can be seen that this approach is inapplicable because the number of false positive is excessive. The third row finally shows the results obtained by the final set of adapted rules.

23.6 Experimental Result

This section presents results obtained by testing the adapted rule set over a real traffic scenario. For simplicity, we have randomly chosen a subset of 1000 Snort rules for hardware implementation over the Snort pre-filter. Experiments have been performed on ten captured traffic traces, amounting to a total of 68 GB of traffic.

In order to assess the effectiveness of the pre-filter, we have compared the following scenarios:

- Unfiltered trace: all the traffic has been directly fed to the Snort software supporting the 1000 rules in their non-adapted (full)version.
- Filtered trace: the traffic has been first delivered to the Snort hardware pre-filter, supporting the 1000 reduced Snort rules. The output of the pre-filter has been then delivered to the Snort software, acting in this configuration as the back-end monitoring application, supporting the same set of rules in their full version.

The first performance metric consists in measuring the ability of the Snort pre-filter to reduce the data delivered to the back-end application. Of the 68 GB traffic, only 4.83 GB of data was delivered; in other words, the Snort pre-filter was able to

Table 23.5 Experimental Result

	Unfiltered trace	Filtered trace
Alert	3579	3540
False negative	0	190
False positive	0	151

filter out 93% of the original traffic. Note that this result was obtained by using a pre-filter implementation which does not restrict to capture only the packets matching a given (relaxed) signature, but further delivers all the *subsequent* packets in the relevant flow.[3]

The second performance metric involves the assessment of the pre-filter effectiveness in terms of false positives and false negatives. Table 23.5 compares the results obtained in the two scenarios. The Snort software operating over the unfiltered traces has revealed 3579 alerts while it has revealed only 3540 alerts over the filtered traces.

The relatively large amount of false negatives obtained (190 undetected attacks over the 3579 ones, according to the unfiltered trace results) is readily explained and imputable to two relatively marginal technical issues to date still affecting our current prototype implementation.

The first issue involves the coding of special characters used in uri-content matching. Our current pre-filter implementation does not yet support the translation of Unicode characters to ASCII which are provided in Snort by the http_inspect preprocessor (whose implementation is out of the scope of our work). As a result, our pre-filter does not match the Unicode representation of the character "/". This clearly emerges, for instance, with the rule identified by the sid 895 (message WEB-CGI redirect access), which produced 97 false negatives since its relaxed version was expected to match the uri-content "/redirect", i.e., including the special character '/'.

The second reason for false negatives is imputable to issues which are not related to our Snort pre-filter, but appear to be related with specific requirements of the legacy Snort *software* in correctly detecting and parsing TCP flows. Indeed, in order to correctly parse a flow, the Snort software requires not only to receive the specific packets matching a specific signature, but it also requires some extra information on the TCP flow itself. To face this issue, we have implemented a software pre-processing module running on the back-end system before the actual Snort legacy software, and devised to "forge" TCP ACK packets and the TCP three way handshake which are filtered out by the HW pre-filter. Our module appears effective in most of the cases, as it provides a

[3] Pre-filter architecture details are out of the scope of this chapter. But, in brief, a filtering table was added to the Snort pre-filter. The table was automatically updated with a flow key extracted from a matching packet, and managed using an LRU (Least Recently Used) policy. All packets whose flow matched an entry of the filtering table were then forwarded. This permits to feed the Snort application operating in the back-end with multiple packets belonging to a same matching flows, and not only isolated matching packets.

formally valid input to the Snort software, and particularly the information needed by the Snort software for determining the state of the session (i.e., if the session is established or not) and the direction of the packets (i.e., from or to server). However, the in-depth analysis of the actual false negatives reveals that even if the relevant signature were correctly captured by the pre-filter and delivered, in some cases, the extra information (ACKs and TCP handshakes) forged by the software pre-processing module were insufficient to drive the Snort legacy software to correctly parse the captured flow.

Finally, note that besides the false negatives detected because of the two above discussed implementation issues, we would have also expected false negatives to occur because of signatures spreading across multiple packets. Indeed, our pre-filter does not perform packet reassembly as this would require to keep per-flow states, and therefore it is vulnerable to signatures split across multiple packets. However, in practice, no such cases have been identified, also because we have employed relaxed rules which are much shorter than the exact rules they were derived from.

Concerning false positives, we have found only 151 cases. A closer scrutiny revealed that 134 out of 151 false positives were due to a single rule, namely sid 8428 (i.e. openssl get shared ciphers overflow attempt), which was undetected in the unfiltered scenario because, in the original trace, the TCP three-way handshake associated to these alerts were not included in the captured traces (i.e., the relevant TCP handshake had happened before the trace capture starting time), whereas in our case the relevant handshakes were properly forged by our software pre-processing module. In other words, these 134 cases should be more properly interpreted as false negatives of the Snort legacy software because of boundary effects of the considered packet trace, rather than false positives of the Snort pre-filter). This leaves an excellent performance of just 17 "true" false positives for our Snort pre-filter operating on normal real world traffic.

23.7 Conclusions

We presented the design and implementation of an hardware-based front-end pre-filter for the topmost known Snort Intrusion Detection System (IDS). Moreover, we have proposed an experimentally-driven heuristic rule adaptation methodology of the Snort rules so that they can be supported over a commercial, low-end, FPGA board, meanwhile providing good filtering performance. In particular, leveraging a large amount of real world traffic traces, we have determined through a cycle of experiments, how these rules can be properly simplified. Finally we have demonstrated that about 1000 adapted Snort rules can be supported over a low-end NetFPGA hardware based Snort pre-filter, with 93% data reduction efficiency, retaining a comparable detection performance with respect to the original, non adapted, rules.

References

1. Sourcefire: Snort: The open source network intrusion detection system. http://www.snort.org (2003)
2. Haoyu Song Sproull, T., Attig, M., Lockwood, J.: Snort offloader: a reconfigurable hardware NIDS filter. In: International Conference on Field Programmable Logic and Applications (2005)
3. Yang, Y.H.E., Jiang, W., Prasanna, V.K.: Compact architecture for high-throughput regular expression matching on FPGA. In: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, pp. 30–39 (2008)
4. Bispo, J., Sourdis, I., Cardoso, J., Vassiliadis, S.: "Synthesis of Regular Expressions Targeting FPGAs: Current Status and Open Issues", Reconfigurable Computing: Architectures, Tools and Applications. Springer
5. Lin, C., Huang, C., Jiang, C., Chang, S.: Optimization of pattern matching circuits for regular expression on FPGA. IEEE Trans. VLSI Syst. **15**(2), 1303–1310 (2007)
6. Moscola, J., Lockwood, J., Loui, R.P., Pachos, M.: Implementation of a content-scanning module for an internet firewall. In: Proceedings of 11th Annual IEEE Symposium Field-Programmable Custom Computing Machines (FCCM '03), pp. 31–38 (2003)
7. Sidhu, R., Prasanna, V.K.: Fast regular expression matching using FPGAs. In: Proceedings of Ninth IEEE Symposium Field-Programmable Custom Computing Machines (FCCM) (2001)
8. Smith, R., Estan, C., Jha, S., Kong, S.: Deflating the big bang: fast and scalable deep packet inspection with extended finite automata. ACM SIGCOMM Comput. Commun. Rev. **38**(4), 207–218 (2008)
9. Baker, Z.K., Prasanna, V.K.: Automatic synthesis of efficient intrusion detection systems on FPGAs. IEEE Trans. Dependable Secur. Comput. **3**(4), 289–300 (2006)
10. Lockwood, J., McKeown, N., Watson, G., Gibb, G., Hartke, P., Naous, J., Raghuraman, R., Luo J.: NetFPGA-an open platform for gigabit-rate network switching and routing. In: IEEE International Conference on Microelectronic Systems Education (2007)
11. Sourdis, I., Dionisios, N., Pnevmatikatos, S.: Scalable multigigabit pattern matching for packet inspection. IEEE Trans. VLSI Syst. **16**(2), 156–166 (2008)
12. Greco, C., Nobile, E., Pontarelli, S., Teofili, S.: An FPGA based architecture for complex rule matching with stateful inspection of multiple TCP connections. Programmable Logic Conference (SPL), 2010 VI Southern, pp. 119–124, 24–26 March 2010
13. Sourdis, I., Dimopoulos, V., Pnevmatikatos, D., Vassiliadis, S.: Packet pre-filtering for network intrusion detection. In: Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems (2006)

Chapter 24
Introducing Privacy Awareness in Network Monitoring Ontologies

Giuseppe Tropea, Georgios V. Lioudakis, Nicola Blefari-Melazzi, Dimitra I. Kaklamani and Iakovos S. Venieris

Abstract The availability of IP traffic monitoring data is of great importance to network operators, researchers and law enforcement agencies. However, privacy legislation, commercial concerns and their implications constitute an impediment in the exploitation of such data. In order to allow compliance to the derived issues and protect privacy without compromising information usability, this chapter leverages findings from two separate research initiatives and aims at paving the way towards a unified approach for privacy-aware collection, processing and exchange of data that stem from network monitoring activities. It investigates the fundamental principles and requirements for a privacy-aware ontological model in the semantic domain of monitoring-data management and exchange, as well as a rule-based approach in specifying the appropriate privacy policies, and enables a clean separation between data models and security semantics. It pursues the definition of the appropriate structures for seamlessly introducing privacy awareness in network monitoring ontologies, including user context, intended usage purpose, data age and privacy

G. Tropea (✉) · N. Blefari-Melazzi
CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni,
UdR Roma Tor Vergata, Rome, Italy
e-mail: giuseppe.tropea@cnit.it

N. Blefari-Melazzi
e-mail: blefari@uniroma2.it

G. V. Lioudakis · D. I. Kaklamani · I. S. Venieris
e-mail: gelioud@icbnet.ntua.gr

D. I. Kaklamani
e-mail: dkaklam@mail.ntua.gr

I. S. Venieris
e-mail: venieris@cs.ntua.gr

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_24,

obligations. Such an approach enables to transfer the expressiveness of legislation rules into the model and allow their automatic processing.

Keywords Network monitoring · Privacy · Ontology

24.1 Network Monitoring and Personal Data Protection

Intuitively, since network monitoring depends by default on the collection and processing of information, it raises issues related to the protection of personal data and is surrounded by legal implications [1]. Furthermore, network-monitoring activities are of peculiar interest, compared to other domains (e.g., e-commerce), as far as privacy protection is concerned, for a number of reasons:

- Privacy-sensitive information is not limited to the payload of the network packets, i.e., the content of the monitored communications. In fact, this case could be even considered as trivial from a privacy protection point of view, since the confidentiality of the content can be adequately achieved by using strong end-to-end encryption. The focus of passive network monitoring is on the collection of so-called context data [2]. In [3], such data are characterized as "semi-active", in the sense that data collection occurs transparently for the user; this type of transparent, implicit collection tends to raise greater privacy concerns than those initiated by the user [4].
- While the various protocols' headers already reveal much information (e.g., a visited web-site or the peers of a VoIP call), a huge amount of personal information can be further extracted from their processing, even if they have been anonymized. Literature has shown that once a flow can be examined in isolation, fingerprinting techniques allow deriving personal information from as little as the basic statistics (i.e., packet sizes and inter-arrival times' correlation) of the delivered packets [5–7]. Moreover, as [8] demonstrated, SSL/TLS does not resist statistical traffic analysis, while meaningful data can be finally extracted even from "unsuspicious" header fields, such as the IP ID alone [9].
- The network monitoring activities, as well as the underlying categories of data, have been subject of specific regulations, such as [10, 11] in Europe and [12] in the USA. Additionally, in many countries, independent protection authorities regulate and audit privacy protection in communications.

Indeed, the legislation plays a crucial role in the determination of data collection and processing policies, in the context of network monitoring; the underlying requirements originate not only from data protection domains (e.g., [13]), but also from provisions related to public welfare, such as public security. The thorough description of the underlying legal framework is beyond the scope of this chapter; the reader is referred to [1, 14]. Drawing from the legal/regulatory, as well as operational aspects of network monitoring, the requirements for a unified approach to privacy preservation shall concern the controlled access to the data,

implemented in a privacy-aware manner and complemented by the enforcement of well-defined strong anonymization patterns. The approach shall be grounded on the semantics of data, monitoring purpose and roles of the involved entities, as well as any other constraint that can be characterized as "privacy context".

The challenge of network traces protection has recently been the focus of several research approaches. Among the most prominent ones, the anonymization API (AAPI) [15], as well as the IETF IPFIX working group's initiative for introducing anonymization support to the IPFIX information model and protocol [16, 17] are included. Nevertheless, such approaches suffer from being based on "static" anonymization policies, while they are vulnerable to attacks able to infer sensitive information [18, 19]. On the other hand, approaches based on differential privacy, such as [20], or mediated trace analysis [21], are effective, but not applicable in real-time applications.

The FP7 projects PRISM [22] and MOMENT [23], despite their differences, have both dealt with unified, privacy-aware access to network data, proposing complementary approaches. The work within MOMENT has focused on a mediator, based on a Service Oriented Architecture (SOA) [24], comprising a unified interface to heterogeneous measurement infrastructures and data repositories. An ontology of the IP traffic measurement domain has been developed, in order for the system to overcome differences between various tools and techniques. The project has deployed a semantic query interface acting as a front-end for the plethora of measurement data sources available nowadays, enabling inference and automatic reasoning capabilities over the retrieved measurement data. On the other hand, PRISM has proposed a general purpose, privacy-preserving passive network monitoring system, with privacy-related functionalities spanning across the whole system, from the traffic probe to the adaptation of monitoring applications. PRISM grounded its access control and authorization infrastructure on an ontology, conceived on the basis of the data protection legislation.

The aim of the work presented here is to pave the way towards a unified approach for privacy-aware collection, processing and exchange of data that stem from network monitoring activities, by contrasting the lessons learned from the MOMENT and PRISM initiatives and providing inputs to other similar ongoing activities. Especially, the work described here contributes to the goals of two initiatives in which the authors are active. On the one hand, the FP7 project DEMONS [25] is presently dealing with similar issues, aiming at privacy-preserving mechanisms for cooperative network monitoring. On the other hand, the "Monitoring Ontology for IP traffic" (MOI) [26] Industrial Specification Group (ISG) of the European Telecommunications Standards Institute (ETSI) deals with the specification of a semantic model for the description of network monitoring information, in order to allow interoperable IP Traffic Monitoring and Analysis techniques and protocols.

The rest of this chapter is organized as follows: Section 24.2 provides an overview of the two projects, MOMENT and PRISM, while Sect. 24.3 discusses the derived requirements for a privacy-aware approach in designing ontologies for the network-monitoring domain. Based on these requirements, the ETSI MOI ISG

considerations for a combined semantic model are outlined in Sect. 24.4. Some concluding remarks are provided in Sect. 24.5.

24.2 Two Perspectives on the IP Traffic Measurement Domain

While MOMENT and PRISM have drawn from different backgrounds and originated from different scientific communities, both projects extensively acknowledged the manifold advantages of using ontologies, which provide a vocabulary of concepts and relations to describe a domain and stress knowledge sharing and knowledge representation [27], in the design of traffic monitoring infrastructures. This section summarizes their main characteristics.

24.2.1 The MOMENT Perspective

Under the umbrella of the MOMENT project, an ontology comprising all aspects of the IP measurement domain has been developed [28, 29]: it includes a Data ontology, a Metadata ontology, an Upper ontology and an Anonymization ontology. The Data ontology describes the hierarchy of the different kinds of measurements and their relations to the physical components of the network; the Metadata ontology describes all the information about how the measurements are stored, how and when they were measured, etc. All concepts common to those ontologies, such as time, units and context, are described in an Upper Ontology and finally, the Anonymization ontology describes dependencies between the possible anonymization strategies that data have to undergo, prior to being released to the user requesting them, and the role and purpose of such a user within the community of people interested in network measurements.

This design allows for information to be placed at different abstraction levels, including the definition of specific class of measurements that are derived from generic ones and introducing the concept of meta-information (so that users can also request a view of what the system knows and what they are allowed to ask).

Several iterations were needed to achieve a generic and powerful enough model, which is able to accommodate for all the schemas contained inside all data repositories that are connected to the MOMENT mediator. First iterations on the ontology were designed based on a strict hierarchy of network measurements categories, onto which many data sources failed in smoothly mapping their data. This approach has changed in later revisions of the semantic model, see [29], and the final Data ontology gives much more importance to the details of the information the measurement carries within itself, rather than trying to assign the `Measurement` class to one of a set of predefined categories under a fixed hierarchy. Specifically, information carried by the measurement is modeled through the `hasMeasurementData` property and the instances of

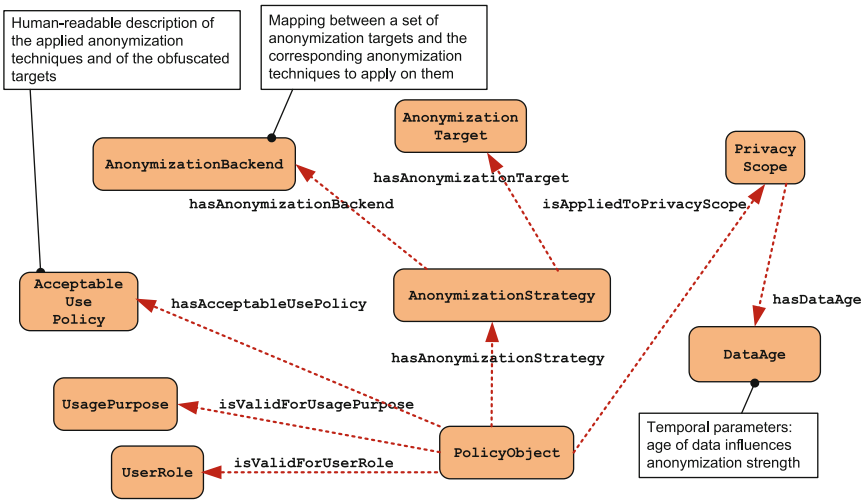


Fig. 24.1 MOMENT Anonymization ontology

`MeasurementData` subclasses, one for every possible measurement value. Other high-level concepts such as `Route`, `Capacity`, etc., which cannot be determined with single values, are represented with the `Metric` class.

Moreover, since its definition, the MOMENT project has also been concerned about obfuscation of certain fields of the data, which are passed to the end-user, in order to enforce a layer of anonymization for protection of the data originator. The `PolicyObject` is the cornerstone of the Anonymization ontology, as shown in Fig. 24.1. It can be viewed, in OWL terms, as an N-ary relation that associates together a number of `UserRoles` and a number of `UsagePurposes`, applied to a number of `PrivacyScopes`. The `PolicyObject` specifies a well-defined `AnonymizationStrategy` and an associated `AcceptableUsePolicy`. The `AnonymizationStrategy` is composed of a group of `AnonymizationTargets` and an `AnonymizationBackend` to support and implement that strategy, i.e., the specific external tool that will ultimately be invoked to do the real anonymization job.

Two innovative ideas are applied to the Anonymization ontology: the Network Data Age and the Acceptable Use Policy. The first technique is employed to capture the common concept that, when a Measurement was generated a long ago, it usually becomes less sensitive, so than a looser anonymization scheme could be enforced. This is captured by the `DataAge` value partition class, together with a dynamic behavior attached to that class, and based on fuzzy membership functions. It provides a convenient way to store the chosen linguistic labels (i.e., OLD, RECENT, NEW) inside the ontology. The fuzzy approach allows bridging of precise numerical values (date/timestamp is converted into number of seconds elapsed since the measure was taken) to all provided linguistic labels (to a different degree with each one of them) via a fuzzy membership function.

The Acceptable Use Policy simply represents an informative document, although structured, about what the provider expects from the user regarding the usage of the data that the provider itself is willing to release. This approach can be regarded as a kind of End User Legal Agreement, but in the field of network measurement. At the time of release of the results to the end-user, the AUP is constructed on-the-fly by the system based on the applied policy and it is given back to the user together with the results.

24.2.2 The PRISM Perspective

Given the importance of semantics in personal data protection, the PRISM ontology [30] has been developed in order to become the semantic implementation of a privacy-aware access control and authorization model [31], specifically devised for the protection of network monitoring data. The PRISM access control mechanism constitutes a two stage approach, where the underlying policies are described by means of semantically defined X.509 Attribute Certificates [32], as far as their static part is concerned, while the dynamic "privacy context" is evaluated in real-time, by means of direct ontological reasoning.

Essentially, the PRISM ontology defines access control rules as associations of data types, monitoring purposes and operational roles. In that respect, the instances of the `Rules` class implement the access control provisions, by connecting instances of the `PersonalData`, `Purposes` and `Roles` classes, respectively. Each of the latter three classes is characterized by a number of OWL object properties, defining different directed graphs over the same sets of instances:

- The instances of the `PersonalData` class are hierarchically organized in three different ways. The `inheritsFromData` object property specifies inheritance of properties, the `lessDetailedThan` and `moreDetailedThan` define detail level transitions, while the `containsType` property puts in place an AND-tree hierarchy.
- Purposes are organized by means of two relations that define OR-tree and AND-tree hierarchies, implemented by the `inheritsFromPurpose` and `consistsOf` properties defined over the instances of the `Purposes` class.
- Similarly, the `inheritsFromRole` and `isPartOfRole` properties define OR-tree and AND-tree hierarchies over the different roles of the system.

The instances defined as members of these three classes have been selected after the elaboration of data models, as well as the actual needs identified in the context of the project and described as requirements. In that respect, the `PersonalData` class contains types coming from the network (IPv4 and IPv6) and transport layers and application-specific data, as well as data types of the IPFIX protocol [16], which is extensively used by the PRISM project. The `Purposes` class includes a variety of monitoring purposes coming from heterogeneous

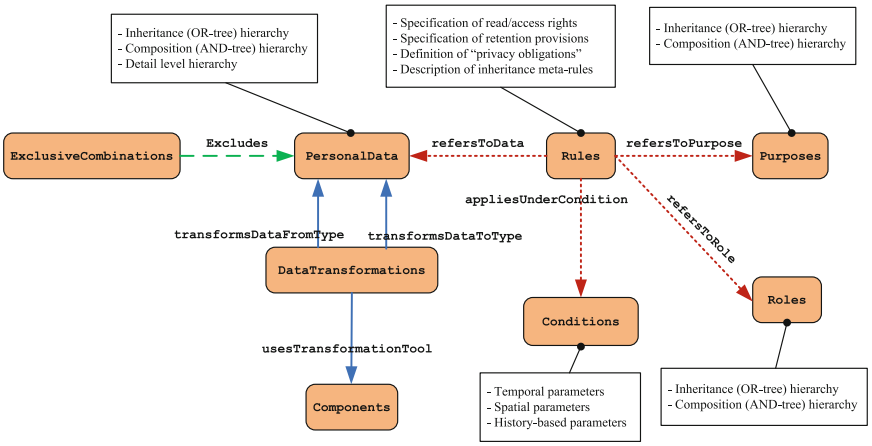


Fig. 24.2 PRISM ontology

domains, while the `Roles` class reflects a simplified, yet realistic, operational structure within a network operator.

Each rule specifies three aspects of data management: read access rights, write access rights and data retention. When the rule is subject to conditional provisions, the `appliesUnderCondition` property links the rule with some instance of the `Conditions` class, which specifies spatial, temporal and history-based conditions for the enforcement of the rule in question. Moreover, each rule defines possible complementary actions that should be executed along with the rule's enforcement, frequently referred to in the literature as "privacy obligations" [33]. Finally, each rule can have meta-rules, reflecting concepts such as inheritance.

The PRISM ontology introduces also the concept of exclusive combinations of data. In this context, the semantic definitions of different data types may be members of relations that are defined as `ExclusiveCombinations` instances and impose restrictions on the disclosure of some data types depending on prior disclosure of other types.

For the specification of anonymization norms, the PRISM ontology incorporates the `DataTransformations` class, the instances of which specify the processing actions that lead from a set of data types to another. Each transformation is linked to an instance of the `Components` class containing the "semantic signatures" of PRISM processing components, being either proprietary Java modules or wrapped functionalities offered by the AAPI [15]. It should be noted here that what this class' instances specify is the capabilities offered by the underlying PRISM systems. The actual transformations that take place comprising the anonymization strategy, are determined in real-time based on the ontological reasoning and take the form of a workflow specification.

While a more detailed description of the PRISM ontology can be found in [30], Fig. 24.2 presents a simplified version, illustrating the most important interconnections between the PRISM ontological entities.

24.3 Requirements for a Unified Model

Each of the models proposed by the PRISM and MOMENT projects was representative of the needs of a set of stakeholders, with the sets having some overlap in interests. Nevertheless, in order to be able to provide a solution that satisfies all stakeholders (telecom companies, researchers, the legislator and law enforcement agencies, as well as the users of the networks) whilst providing necessary levels of security and accessibility to the measurement data, a higher-level approach in the definition of the requirements supporting privacy-awareness via the various concepts and techniques involved is necessary.

Semantics of the Networking-Measurements Domain

First of all, an integrated approach shall both tackle the privacy-protection issues involved in any activity of IP traffic collection, storage, analysis and, within the same design, ease the access to the available repositories of IP measurements, helping to solve interoperability issues and the formulation of complex, semantically-enriched queries to the various distributed databases.

This means that the model should be equally focused on data processing and on data sharing operations. Thus, the ontology must not lack support for measurements specifically focused on targets which do not fall into the "flow" paradigm, meaning, for example, that a graph for concepts such as link bandwidth or capacity must be directly supported, since it evades the classic four-tuple {sourceIP, sourcePORT, destIP, destPORT} paradigm. This allows the model to manage measurements which are part of self-contained experiments and form collection of packet traces, possibly already stored in internal data bases, rather than only serving the purpose of observation and filtering of real-time traffic flows.

On the other hand, the conceptual model must not lack the high-level concept of a "flow", that is an end-to-end long term packet stream, and be able to enforce analysis patterns and security procedures in such cases as the capture of an SSH session or of an HTTP download between two hosts.

Interoperability and Communications Paradigm

Mechanisms for data collection and management in the realm of IP measurements, meet more and more the *Data Exchange* paradigm as opposed to the *Data Publishing*. In the latter, the owner of the measurements database can publish the data so that they are available to the community and the other stakeholders, whereas in the case of Data Exchange the (even real-time generated) data are exchanged with other peers, meaning also the possible back-office systems of the own organization. Under this paradigm, data querying capabilities are to be ensured and the role of the security and privacy ontology is to reveal to the involved parties only a controlled sub-set of data. Thus, the model should be able to capture concepts such as:

- The peers of the data exchange, along with the means for their authentication and authorization.
- The underlying purpose and intended use of the data.
- Information, consent and any other action to accompany the exchange.
- Data retention periods and the associated provisions for their deletion.
- Level of trust of the data.
- Information about quality, location, creation of the data.

The decision process must be enabled to take such information into account when defining the sub-set of data to disclose to the requesting party.

Separation Between Model and Rules

The set of rules needed to control the sub-set of data to release or obfuscate must abstract the complexity of privacy requirements [14], in order to facilitate its selection, automation and enforcement at the process level. Within the term "privacy requirements" all obligations involved in the process of sharing and processing the data have to be considered. A fundamental requirement for this to be accomplished is that the security and privacy model should be neatly decoupled from the model of the network-monitoring data it operates on, meaning that the design must decouple the privacy aspects from their respective data models.

Moreover, since legal rules are usually expressed in the form of if–then clauses, relationships and implications about privacy and security that are hard to be expressed in OWL due to its limited expressiveness for describing such properties, need to be modeled by adopting a notation to express the privacy constraints similar to the Semantic Web Rule Language (SWRL) [34]. The adoption of SWRL and the associated Semantic Query-Enhanced Web Rule Language (SQWRL) [35] query language, allows for direct incorporation of OWL constructs inside the specification of rules, by maintaining a clean separation between rules and model. This approach allows for seamless representation of a privacy policy, stating for example that "link delay measures from Italian telecom operators sent to research institutions must be accompanied by a non-disclosure agreement", without any need to clutter the data model with classes representing the concept of "non-disclosure agreement". This additionally permits customization of privacy policies to specific environments or for specific purposes, without the need to restructure the data model or even to fully understand its details.

Implications for the Privacy Due to the Architecture of the Data Model

A clear distinction between schema (classes) and data (individuals) should be maintained. It allows distinguishing between model developers and model users. Up to now, the adoption of a class-only paradigm for modeling a complex domain has suffered from the availability of the SPARQL query language [36] only, which struggles in when the simple and elegant triple-based RDF representation of the OWL model is obfuscated by class-only property restrictions. The availability of

SQWRL changes this scenario. Under this respect, privacy-awareness in the networking measurements domain is quite challenging, since it is quite normal to deal with terabytes of data that we would like to naturally fit as OWL individuals. Although a raw import of measurement data into the ontology for privacy-compliant processing is still to come, a clean model supporting this view by means of standard tools is the correct approach, together with aggregation techniques that reduce the computational burden. Proprietary tools running outside (i.e., executing workflows on the packet traces) would need to comply with the inferred policies.

24.4 Combining Semantic Models

Keeping all requirements discussed in Sect. 24.3 in mind, this section leverages the experience gained through MOMENT and PRISM projects, and describes considerations for a unified model which shall cover all aspects of traffic monitoring, including the semantics for a privacy-preserving handling of measurement data. Table 24.1 summarizes the main points of MOMENT and PRISM, which are the starting point of the ETSI MOI ISG. This initiative has already performed the first steps towards a combined semantic model.

Something that immediately differentiates the two approaches concerns their overall structure. On the one hand, MOMENT defines four different ontologies in contrast to PRISM, which adopts an integrated approach; on the other hand, the classic "classes vs. instances" [37] design strategy dilemma further differentiates the two projects. These two fundamental differences originate from the purpose and scope of ontologies' use in the context of these projects.

PRISM uses the ontology as the very detailed privacy configuration of its underlying systems and all decisions are highly dynamic in nature, especially since they apply not only to data stored in repositories, but also to data collected in real-time. In that respect, dynamic data processing workflows are being specified on-the-fly, taking into account all the underlying parameters; reasoning in PRISM relies on a variety of attributes, and the "classes approach" together with a legacy reasoner are unable to return the desired results. Therefore, and in order to achieve the desired flexibility, PRISM has preferred heavy use of instances and the proprietary PRISM reasoner operates on several complex graphs that are defined over the instances clouds.

On the other hand, MOMENT makes use of ontologies the "classical" way. While interoperability has not been an issue for PRISM, in MOMENT it plays a very fundamental role and the design adopted is based on that. In fact, MOMENT fosters interoperability between its mediator and a variety of data sources and the ontological model aims at being the enabler for the seamless operation of the underlying SOA-based systems. In order to be general enough and serve the purpose of being a semantic vocabulary between heterogeneous systems, the privacy-awareness-related part of the model is separated from the specific way IP measurement data are modeled. In order to avoid rewriting the Anonymization

Table 24.1 Summary of the MOMENT and PRISM approaches

Aspect	MOMENT perspective	PRISM perspective
General structure	4 discrete ontologies (Data, Metadata, Upper, Anonymization), classes-based	Integrated ontology, instances-based
Data types representation	Sub-classes of the `AnonymizationTarget` class	Instances of the `PersonalData` class, with three different hierarchies
Purposes representation	Sub-classes of the `UsagePurpose` class	Instances of the `Purposes` class, with two different hierarchies (AND-tree and OR-tree)
Roles representation	Sub-classes of the `UserRole` class	Instances of the `Roles` class, with two hierarchies (AND-tree and OR-tree)
Rules definition	Rules serve for the association of anonymization strategies to roles, purposes and anonymization targets	Access control rules, with additional provisions regarding conditions, obligations
Anonymization strategy definition	Static; the Anonymization Ontology defines some pre-specified anonymization strategies	Dynamic; the anonymization strategies are specified in real-time by means of ontological reasoning and taking into account all the contextual parameters that apply
Anonymization implementation	The Anonymization module calls external anonymization back-ends, with AAPI [15] wrapper included	The ontology contains "semantic pointers" to PRISM software components (incl. AAPI wrapper)
Privacy obligations	N/A	Per rule definition, using explicit properties
Exclusive data disclosure	N/A	Implemented by means of the `ExclusiveCombinations` class
Additional "privacy context"	Data age	Temporal, spatial and history-based restrictions, exclusive data combinations
Management of measurements	Measurements are represented as instances of the `Measurement` class	Measurements are managed outside the ontology

model each time the Data model changes, as well as to achieve a perfect decoupling between the description of how IP traffic measurements have to be obfuscated and the description of IP traffic measurements themselves, the Anonymization Ontology avoids embedding any direct reference to the specific Data Ontology classes; the necessary mapping of the classes from the privacy-aware part to the measurements part is done by some runtime Java code.

While each approach perfectly achieved its goals, when having in mind a unified approach, the features of each of the two may become either advantages or disadvantages and, therefore, a unified proposal must exploit certain aspects of each model. With interoperability being the bottom-line requirement for

standardization, we consider here the MOMENT approach as the basis of the unified approach and investigate the additional features that should be adopted with respect to the PRISM approach; the major point for their conceptual integration is the MOMENT Anonymization Ontology.

Concerning the representation of data types, monitoring services and purposes, as well as the roles of the involved entities, the MOMENT Anonymization Ontology already defines such entities; in order to become PRISM-enabled, the corresponding classes (`AnonymizationTarget`, `UsagePurpose` and `UserRole`) can be replaced by the equivalent PRISM classes, which should though be relieved from their interoperability restrictions. In that respect, these structures should be extended by the flexible string matching approach used in the MOMENT model. This way, the unified approach takes advantage of both the detailed hierarchical relationships defined by PRISM and the interoperability and flexibility features of MOMENT.

With respect to the definition of rules, a merge of the two approaches' concepts is deemed necessary. PRISM suffers from the lack of concrete pre-defined anonymization patterns, while MOMENT needs to be extended for supporting real-time strategies specification. Therefore, we envisage defining rules that can be used for real-time decision making, like in PRISM, but expressed in SWRL and directly interconnected to the MOMENT anonymization strategies. In addition, these strategies should be extended with support for privacy obligations, with the latter becoming actually an integral part of the strategies. It is noted that there is a great similarity of the two projects regarding the specification of the software functionalities that implement anonymization; the pre-defined vs. dynamic strategies difference does not have any essential effect on this.

On the other hand, an interesting aspect of the combined approach is the so-called "privacy context", since the two projects have followed quite different approaches. The unified model should provide support for the temporal, spatial, history-based and exclusive data combinations of PRISM, as well as the MOMENT mechanism for taking into account the age of the data, which applies to stored records. We propose a class similar to the PRISM's `Conditions`, which will additionally include the MOMENT's provisions regarding the age of the data. Such a class should be related with both the rules themselves, as well as the anonymization strategies, for their conditional application, while it should also be supported by the MOMENT fuzzy matching mechanism for the interpretation of "linguistic" categories, e.g., "recent measurement" with the numerical values (usually a date in MM:DD:YY:HH:MM:SS format) inside the rows representing the measurements, coming from the databases.

24.5 Conclusions

Standardizing network measurement parameters, algorithms and protocols is the basis of any fair service level agreement and interoperable network management,

as well as cooperative network monitoring. Only very recently the network monitoring community is embracing the concept that this standardization activity cannot neglect privacy concerns. When privacy-preserving techniques are applied to the network monitoring domain, several challenges arise which this chapter outlines, and an integrated approach must be devised based on a combined semantic model enhanced with reasoning functionalities that adequately cover the specific domain. In this context and starting from the mature work of two EU projects, this chapter has investigated the similarities and differences of their semantic models, trying to come up with general requirements and design patterns for a privacy-aware approach to network measurement. Apart from being incorporated in the DEMONS project [25] work, the findings of this chapter will ultimately help the achievement of the ETSI MOI ISG [26] goals.

References

1. Ohm, P., Sicker, D., Grunwald, D.: Legal issues surrounding monitoring during network research, In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07), San Diego, USA, October 24–26, pp. 141–148 (2007)
2. Zugenmaier, A., Claessens, J.: Privacy in electronic communications. In: Douligeris, C., Serpanos, D.N. (eds.) Network Security: Current Status and Future Directions, pp. 419–440. Wiley-Interscience, Hoboken (2007)
3. Lioudakis, G.V., Koutsoloukas, E.A., Dellas, N., Tselikas, N., Kapellaki, S., Prezerakos, G.N., Kaklamani, D.I., Venieris, I.S.: A middleware architecture for privacy protection. Comput. Netw. **51**(16), 4679–4696 (2007)
4. Cranor, L.F.: I didn't buy it for myself. In: Karat, C.-M., Blom, J.O., Karat, J. (eds.) Designing Personalized User Experiences in E-Commerce, pp. 57–73. Kluwer, Norwell (2004)
5. Bissias, G.D., Liberatore, M., Jensen, D., Levine, B.N.: Privacy vulnerabilities in encrypted http streams. In: Proceedings of the 5th Workshop on Privacy Enhancing Technologies (PET 2005), Cavtat, Croatia, May 30–June 1, LNCS 3856 (2005)
6. Crotti, M., Gringoli, F., Pelosato, P., Salgarelli, L.: A statistical approach to IP-level classification of network traffic. In: Proceedings of the IEEE International Conference on Communications (ICC) 2006, Istanbul, Turkey, June 11–15, 2006
7. Hintz, A.: Fingerprinting websites using traffic analysis. In: Proceedings of the 2nd Workshop on Privacy Enhancing Technologies (PET 2002), San Francisco, CA, USA, April 14–15, LNCS 2482 (2002)
8. Sun, Q., Simon, D.R., Wang, Y.-M., Russell, W., Padmanabhan, V.N., Qiu, L.: Statistical identification of encrypted web browsing traffic. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy (SP' 02), Marseille, France, May 12–15, 2002
9. Bellovin, S.: A technique for counting NATted hosts. In: Proceedings of the 2nd ACM Workshop on Internet Measurement (IMW' 02), Berkeley, CA, USA, November 6–8, 2002
10. European Parliament and Council: Directive 2002/58/EC of the European parliament and of the council concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications). Off. J. Eur. Communities L **201**, 37–47 (2002)
11. European Parliament and Council: Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public

communications networks and amending Directive 2002/58/EC. Off. J. Eur. Communities L **105**, 54–63 (2006)
12. United States Code 18, § 2701: Unlawful access to stored communications
13. European Parliament and Council: Directive 95/46/EC of the European parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Off. J. Eur. Communities L **281**, 31–50 (1995)
14. Lioudakis, G.V., Gaudino, F., Boschi, E., Bianchi, G., Kaklamani, D.I., Venieris, I.S.: Legislation-aware privacy protection in passive network monitoring. In: Portela, I.M., Cruz-Cunha, M.M. (eds.) Information Communication Technology Law, Protection and Access Rights: Global Approaches and Issues. IGI Global, Hershey (2010)
15. Koukis, D., Antonatos, S., Antoniades, D., Trimintzios, P., Markatos, E.P.: "A generic anonymization framework for network traffic. In: Proceedings of the IEEE International Conference on Communications 2006 (ICC 2006), Istanbul, Turkey, June 11–15, 2006
16. Claise, B. (ed.): Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. In: IETF RFC 5101, January 2008
17. Boschi, E., Trammel, B.: IP flow anonymisation support. IETF Internet Draft (2009). http://www.ietf.org/id/draft-ietf-ipfix-anon-01.txt
18. Pang, R., Allman, M., Paxson, V., Lee, J.: The devil and packet trace anonymization. ACM Comput. Commun. Rev. **36**(1), 29–38 (2006)
19. Burkhart, M., Schatzmann, D., Trammell, B., Boschi, E., Plattner, B.: The role of network trace anonymization under attack. Comput. Commun. Rev. **40**(1), 5–11 (2010)
20. McSherry, F., Mahajan, R.: Differentially-private network trace analysis. In: Proceedings of the ACM SIGCOMM 2010, New Delhi, India, August 30–September 03, 2010
21. Mittal, P., Paxson, V., Summer, R., Winterrowd, M.: Securing mediated trace access using black-box permutation analysis. In: Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets 2009), New York, USA, October 22–23, 2009
22. FP7 ICT Project PRISM (PRIvacy-aware Secure Monitoring), Home Page: http://fp7-prism.eu/
23. FP7 ICT Project MOMENT (Monitoring and Measurement in the Next Generation Technologies), Home Page: http://fp7-moment.eu/
24. Papazoglou, M.P., van den Heuvel, W.-J.: Service oriented architectures: approaches, technologies and research issues. VLDB J. **16**(3), 389–425 (2007)
25. FP7 ICT Project DEMONS (DEcentralized, Cooperative, and Privacy-Preserving MONitoring for Trustworthiness), Home Page: http://fp7-demons.eu/
26. ETSI Industry Specification Group on "Measurement Ontology for IP Traffic" (ETSI ISG MOI), Home Page: http://portal.etsi.org/MOI/
27. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
28. Tropea, G., Scibilia, F., Blefari-Melazzi, N.: A semantic framework to anonymize network data and define their acceptable use. In: Proceedings of the 18th ICT Mobile & Wireless Communications Summit 2009, Santander, Spain, June 10–12, 2009
29. Salvador, A., López de Vergara, J.E., Tropea, G., Blefari-Melazzi, N., Ferreiro, Á., Katsu, Á.: A semantically distributed approach to map IP traffic measurements to a standardized ontology. IRCC IJCNC Int. J. Comput. Netw. Commun. **2**(1), 13–31 (2010)
30. Lioudakis, G.V., Gogoulos, F., Antonakopoulou, A., Kaklamani, D.I., Venieris, I.S.: Privacy protection in passive network monitoring: an access control approach. In: Proceedings of the 23rd IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA-09), Bradford, UK, May 26–29, 2009
31. Gogoulos, F., Antonakopoulou, A., Lioudakis, G.V., Mousas, A., Kaklamani, D.I., Venieris, I.S.: Privacy-aware access control and authorization in passive network monitoring infra-structures. In: Proceedings of the 3rd IEEE International Symposium on Trust, Security and Privacy for Emerging Applications (TSP-10), Bradford, UK, June 29–July 1, 2010
32. International Telecommunication Union (ITU)—Telecommunication Standardization Sector: Information technology—open systems interconnection—the directory: public-key and attribute certificate frameworks. ITU-T Recommendation X.509, August 2005

33. Casassa Mont, M.: Dealing with privacy obligations: important aspects and technical approaches. In: Proceedings of the International Workshop on Trust and Privacy in Digital Business (TrustBus 2004), Zaragoza, Spain, August 30–September 3, 2004
34. Parsia, B., Sirin, E., Grau, B.C., Ruckhaus, E., Hewlett, D.: Cautiously approaching SWRL. Technical Report, University of Maryland (2005)
35. O'Connor, M.J., Das, A.K.: SQWRL: a query language for OWL. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23–24, 2009
36. SPARQL Query Language for RDF, W3C Recommendation. http://www.w3.org/TR/rdf-sparql-query/, January 2008
37. Samwald, M.: Classes versus individuals: fundamental design issues for ontologies on the biomedical semantic web. In: Proceedings of the European Federation for Medical Informatics, Special Topic Conference, Timisoara, Romania, April 6–8, 2006

Part VI
Content

Chapter 25
Rights Management in Architectures for Distributed Multimedia Content Applications

Jaime Delgado, Víctor Torres, Silvia Llorente and Eva Rodríguez

Abstract There are several initiatives working in the definition and implementation of distributed architectures that enable the development of distributed multimedia applications on top of them, while offering Digital Rights Management (DRM) features. In this chapter, the main features of the MPEG Extensible Middleware (MXM) (ISO/IEC, http://mxm.wg11.sc29.org, 2010) and the new Advanced IPTV terminal (AIT) (ISO/IEC, ISO/IEC JTC1 SC29/WG11 N11230, 2010), which together will form a future second edition of the MXM standard (renamed to Multimedia Service Platform Technologies) are presented. On the other hand, the DMAG's (Distributed Multimedia Applications Group (DMAG), http://dmag.ac.upc.edu, 2010) Multimedia Information Protection and Management System (MIPAMS) (Torres et al., Springer, Heidelberg, 2006) is also presented, highlighting the common ground and differences between MIPAMS and the standards. A set of usage scenarios is also proposed to show how MIPAMS enables the development of applications, on top of it, which deal with the needs of content creators, distributors and consumers according to different business models.

Keywords Digital Rights Management · Business Scenarios · Standardisation · Content Management Architectures · Multimedia Applications

25.1 Introduction

Three standard or standard-based initiatives need to be taken into consideration when dealing with multimedia applications based on Digital Rights Management (DRM) technology. On one hand, the MPEG Extensible Middleware (MXM) [1]

J. Delgado (✉), V. Torres, S. Llorente and E. Rodríguez
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: jaime.delgado@ac.upc.edu

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_25,

and the Advanced IPTV terminal (AIT) [2], which will merge into the second edition of the MXM standard, renamed to Multimedia Service Platform Technologies (MSPT), are the main initiatives from standardization bodies. On the other hand, the DMAG's [3] Multimedia Information Protection and Management System (MIPAMS) [4] is a relevant standards-based architecture implemented as the basis to develop further applications. MIPAMS is not the sole architecture providing DRM features (refer to [5] for a survey), but we will focus on it in relation to the mentioned MPEG standards.

Most of literature refers to DRM as a means to restrict what users can do with content but, in fact DRM can be used in other contexts. However, we have identified several scenarios where DRM architectures enable the development of alternative applications on top of them, as detailed next.

Content registration, protection, search, licensing and access control. This scenario covers a system with full functionality, including content registration and protection; it offers publication, content search, purchase and licensing, authorization and access control. In this case, we first need an interface for content creators to register and publish their content and determine and modify their offers. This functionality is provided by means of specific user applications for editing or otherwise integrated in a web portal. In this scenario, once content is registered, it can be linked from external sites to license it through the mentioned portal, which means that the content promoted in external sites can include specific links towards the licensing portal. Moreover, apart from being linked from other sites, the portal itself would also be useful for promotion.

Content registration and licensing. This scenario is applicable to those cases where there are well established sites that deal with the promotion and collection of content, but for which licensing is not a part of their business model (e.g. Flickr, Picassa, Panoramio, Youtube, etc.). Although content can be directly accessed from those sites, it may be distributed under some restrictions that do not enable users to use it for free. This is the case when content is distributed e.g. under copyright ("all rights reserved") or Creative Commons Non-Commercial models [6]. In this scenario, the DRM architecture could be used to implement a trading portal, devised for formalizing the rights acquisition for personal or professional usage. Content owners or rights holders are responsible for registering content in the trading portal and providing the link towards it. As in the previous scenario, content can be linked from external sites.

Content registration, licensing and access control without protection. This scenario extends the previous one by adding the access to content after the license purchase, which would be authorization-based, but giving the unprotected content to the purchasing user so that they can enjoy it without further DRM restrictions.

Content registration, search, licensing and access control without content management. This scenario is devised for content providers or distributors that want to use their specific protection mechanisms and content management systems so that content does never leave their well-established systems. In such scenario, when registering content in the DRM Architecture, specific proprietary identifiers

are used for identifying external content. Once objects are registered, rights offers can be published and licenses issued without any technical restriction. Content providers or distributors will have to design their own applications that access content, such as players and editors, in order to manage the access to encryption keys and content from their systems, or otherwise provide an API so that their content can be accessed from third-party applications.

Content registration, protection, search, licensing and access control for limited capabilities devices. Since this scenario involves limited capabilities devices, in some cases, the encryption strength being used should be limited so as not to be detrimental to the devices performance.

Content registration and licensing through external services. This last proposed scenario is based on the usage of registration functionalities, leaving content licensing for being tackled by external sites or services. In this scenario, the DRM architecture should act as a mere intellectual property registry, proving content ownership and offering the possibility to link content with external sites that deal with its commercialization.

In subsequent sections, after describing MIPAMS, MXM and AIT, we will illustrate how MIPAMS has been used to implement these scenarios.

25.2 The DMAG's MIPAMS Architecture

MIPAMS (Multimedia Information Protection and Management System) [4] is a service-oriented DRM platform developed by the DMAG (Distributed Multimedia Applications Group) [3].

The MIPAMS architecture is based on the flexible web services approach, as it consists of several modules and services, which provide a subset of the whole system functionality needed for governing and protecting multimedia content. One of the advantages of having service-oriented DRM functionality relies on the possibility of decoupling it into different subsystems depending on the needs of the application that is going to be implemented, while being able to share the same common services between different applications with different requirements, thus reducing costs. MIPAMS encompasses an important part of the content value chain, from content creation and distribution to its consumption by final users.

Figure 25.1 depicts the MIPAMS architecture, for which we provide next a general overview of its components and the different services being offered.

The Content Service (CS) enables applications to upload and download digital resources such as audio or video files, text documents, etc. Those resources can be optionally encrypted under request, according to the available encryption mechanisms it provides. If encryption is selected, the protection keys will be first requested to the Protection Service and then registered through the same service, once encryption is performed. Content upload requires content to be uniquely identified. Since MIPAMS deals with single resource objects, the identifier being

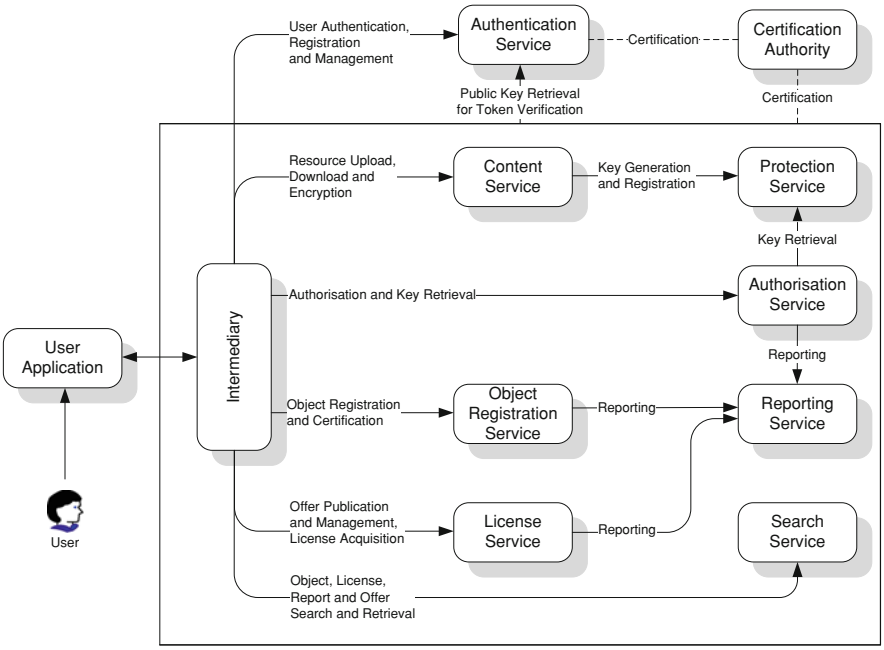


Fig. 25.1 DMAG-MIPAMS architecture

associated to content will be the same one used for the object that contains it, and must be passed as input argument. This identifier can be requested to the Object Registration Service prior to the content upload, or obtained from an external application using MIPAMS (it depends on the scenario).

The Object Registration Service (ORS) enables applications to request a digital representation of content and metadata (i.e. digital objects) to be generated and registered in the system. Content and metadata are packaged together following the MPEG-21 Digital Item [7] approach. Once registered, objects are digitally signed by the ORS so that they can be checked for authenticity and integrity. The ORS also provides unique identifiers for those applications that need to upload content to the CS, as already explained.

The License Service (LS) deals with rights offers and the issuance of licenses. Rights offers are set up by content creators or rights holders after registering content. They include the rights being offered for acquisition by other users and the conditions being applicable to those rights. License issuance refers to the process by which a license is generated as the result of a rights purchase, acquisition or because a rights holder directly grants some user a set of rights. Licenses are expressed using MPEG-21 Rights Expression Language [8].

The Authorization Service (AS) checks whether a user owns any appropriate license that grants him the right to perform a requested action (e.g. play) over a

digital object. The authorization is based on the mechanism defined in [8]. The AS shares the access to the license repository with the LS. If the user is able to perform the action and the requested content is encrypted, the AS will retrieve the encryption keys from the Protection Service and return them to the requesting application. This is the only means for accessing encryption keys, which is performed as an atomic operation.

The Protection Service (PS), as introduced before, generates encryption keys upon request, registers encryption keys associated to uniquely identified content and provides the encryption keys for protected content to the AS. When using MPEG-21 Intellectual Property Management and Protection [9] scheme and descriptors, the PS also offers the possibility to download the protection tools being used by those applications that might be out-of-date.

The User Application (UA) is the player, edition tool, browser or any other means that is managed by the user to deal with the DRM functionality, such as registering and accessing protected contents. The UA may have an internal trusted module or intermediary to enforce DRM, which could consist of a secure local repository for licenses, protection information, offline operation reports and other critical data. In those cases, it may be responsible for estimating tool fingerprints, require offline authorizations, unprotect content, track offline operations and manage content protection information.

The Intermediary may be an integral part of the UA or otherwise be located in the server part (e.g. web portal, brokerage service) to reduce the UA complexity. It can be seen as a broker to whom the UA requests different operations to be performed, as object registration, content upload, rights offer management, license acquisition, authorization, etc.

The Search Service (SS) enables applications to perform accurate searches amongst metadata in the MIPAMS system. That is, it is the front-end for requesting any information present in MIPAMS services databases. Thus, it can be used for searching content, licenses, offers or reports or a combination of them.

The Reporting Service (RS) collects usage reports regarding the registration of objects, the issuance of licenses and the authorizations being performed. It is also capable of building standards-based representations of those reports, such as MPEG-21 Event Reports [10]. Those reports may be used for computing statistics as well as for billing or tracking purposes.

The Authentication Service (ATS) is needed to authenticate the identity of users. It generates SAML (Security Assertion Markup Language [11])-based tokens that identify MIPAMS users. Any service in the MIPAMS architecture will require a token argument to be provided in order to authenticate users. Tokens are digitally signed by the ATS, so that they can be checked for authenticity and integrity by the receiving service. Moreover, the ATS deals with user registration and management (i.e. personal data modification, user account deactivation, etc.).

Finally, there is a need for having a recognized Certification Authority (CA), which issues credentials for the different Components and Actors in the system, as X.509 certificates and private keys for the different architectural components.

25.3 Standard Architectures: MXM and AIT Towards MSPT

The MPEG Extensible Middleware (MXM) [1] is an initiative of the MPEG standardisation group (ISO/IEC JTC1 SC29/WG11). This standard specification defines a middleware platform and a complete set of APIs and protocols for the management of digital content. It promotes the reusability of the MPEG technologies that provide the required functionalities to interoperable Digital Rights Management architectures such as that described in [12]. The MXM standard comprises four public documents, which include the MXM architecture and technologies [13], MXM application programming interface [14], MXM reference software [15] and MXM protocols [16].

The relationship between MIPAMS and MXM is mainly related to some of the protocols defined by MXM. In fact, some MIPAMS services (ORS, CS, RS and LS) follow and implement the concepts behind MXM protocols related to Content, Licenses and Event Reports. MIPAMS also implements part of the MXM engines (Digital Item Engine, REL Engine, etc.), but it does not implement any functionality for modifying digital resources (image, video, audio, graphic, etc.) as it works at a higher abstraction level, the one described by MPEG-21. As digital resource operations are not implemented, Digital Item Adaptation operations considered in MXM have been discarded in current version of MIPAMS. In previous versions of our architecture, we considered MPEG-21 Digital Item Adaptation [17] operations but only to include MPEG-21 DIA expressions into MPEG-21 REL [8] licenses for authorizing complex digital resource adaptations to be rendered in limited capabilities devices (e.g. mobile devices) [18].

The Content Search and Security Engines described by MXM are also considered into MIPAMS Search Service and Protection Service, respectively. Regarding MPEG-21 Media Value Chain Ontology (MVCO) [19], we are implementing an authorization engine based on this ontology to be integrated in MIPAMS.

The authorization engine, either REL-based or MVCO-based, depends on the needs of the final user application using MIPAMS. To sum up, MIPAMS implements most of the engines and protocols described in MXM, although we have used a different approach in some cases. Regarding the functionality defined in MXM that is not part of MIPAMS, we are going to implement MVCO-based authorizer, but we do not plan to include low level resource (audio, images, etc.) operations for the moment.

On the other hand, the Advanced IPTV (AIT) [2] is a joint initiative of MPEG and ITU-T SG16. These two groups started together the standardisation of a set of protocols and interfaces to enable new multimedia services in different environments, for example broadcasting. The requirements for an AIT terminal were defined [20] in conjunction with a set of candidate existing technologies that satisfied some of them. Then, a Call for Proposals was issued in January 2010 and the responses, fulfilling some of the requirements, were evaluated during the 92nd MPEG meeting (April 2010). Since some of the requirements had not received enough contributions, a second Call for Proposals [21] was launched.

Responses with relevant technologies were evaluated [22] during the 93rd MPEG meeting, in July 2010. It was decided that the selected technologies should be integrated with the existing MXM specification resulting in a second edition of MXM, to be renamed to Multimedia Service Platform Technologies. This second edition is now in an initial stage. For example, Working Drafts have been produced for the part on Basic Services [23], which initially was only dealing with the MXM protocols, and the new part on Service Aggregation [24], still in a very early stage.

Although, as already mentioned, AIT is still in a very early stage of development, we are already in a position to compare it with MIPAMS. There is a close relationship between MIPAMS and AIT, as MIPAMS already implements several services described as AIT Elementary Services in [2], especially those related to content, licenses and event reports. MIPAMS also implements several services related to users and services, mainly the ones related to authentication and authorization.

Regarding contracts, we do not implement any functionality inside MIPAMS current version, but we have some background on this area developed during the AXMEDIS project [25], where we analyzed several contracts from partners related to the audiovisual environment and we implemented an application for transforming textual contracts into MPEG-21 REL licenses. Our experience in this field has been reflected in a contribution to the AIT first call for proposals for the elementary services related to contracts [26]. We improved this contribution for the second call for proposals [21] and, as a result of this, a new part of MPEG-21 (part 20), called Contract Expression Language [27], has been launched.

On the other hand, MIPAMS does not include the implementation of the AIT elementary services related to devices, since this functionality is not required in our current scenarios. However, in other contexts, we have defined and implemented operations over devices, as in the AXMEDIS project (see Sect. 25.4.5), but we decided not to include them on MIPAMS as they eventually depend on the final user application (or in the intermediary) or they may change from application to application and need to be implemented according to the application requirements. Nevertheless, we are considering the specification of AIT devices functionality in a specific new module or as an extension of the ATS service.

Regarding services related to groups, in the IPOS project [28] (see Sect. 25.4.3) we have implemented grouping functionality in order to permit users to act as a single unit, while being flexible enough to support different shares. This specific part of IPOS can be easily integrated into the MIPAMS architecture by extending the authentication service, whenever it is needed.

Finally, it must be noted that the first implementations of MIPAMS were made before the start of work on MXM and AIT [29].

25.4 Digital Rights Management Integration

In this section we present the results of some research and development projects where we have implemented the usage scenarios identified in the introduction (see Sect. 25.1) using MIPAMS services and modules.

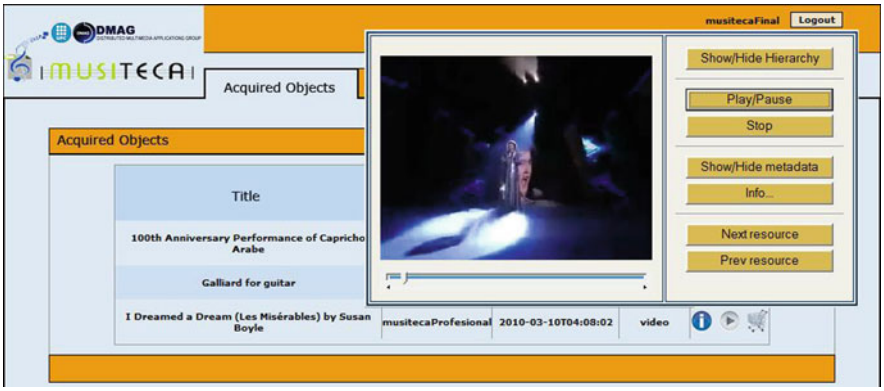


Fig. 25.2 Protected rendering in a specific DRM portal in Musiteca

25.4.1 Content Registration, Protection, Search, Licensing and Access Control

This scenario has been implemented in Musiteca [30], a research project funded by the Spanish Administration. In this project, we have used some of the services conforming MIPAMS (LS, RS, ATS, CS, ORS, SS and CA) to implement a content creation platform. The access to the Musiteca repository is done through a portal that enables the registration of content, the purchase of content (licensing) and the access to the content purchased after the user is authorized. The actions performed by the different users of the system are registered using the reporting functionality provided by RS. Figure 25.2 shows a screenshot of the portal, where content is being rendered.

25.4.2 Content Registration and Licensing

This scenario has been also implemented in Musiteca [30]. Figure 25.3 shows how content is linked from an external site, the Musiteca Freebase database [31], which holds information about musical content in the Musiteca project, to a specific trading or licensing portal.

25.4.3 Content Registration, Licensing and Access Control without Protection

This scenario has been implemented in the Intellectual Property Operations System (IPOS) [28], a Content Management System (CMS) resulting from several software developments done by the DMAG under different contracts with

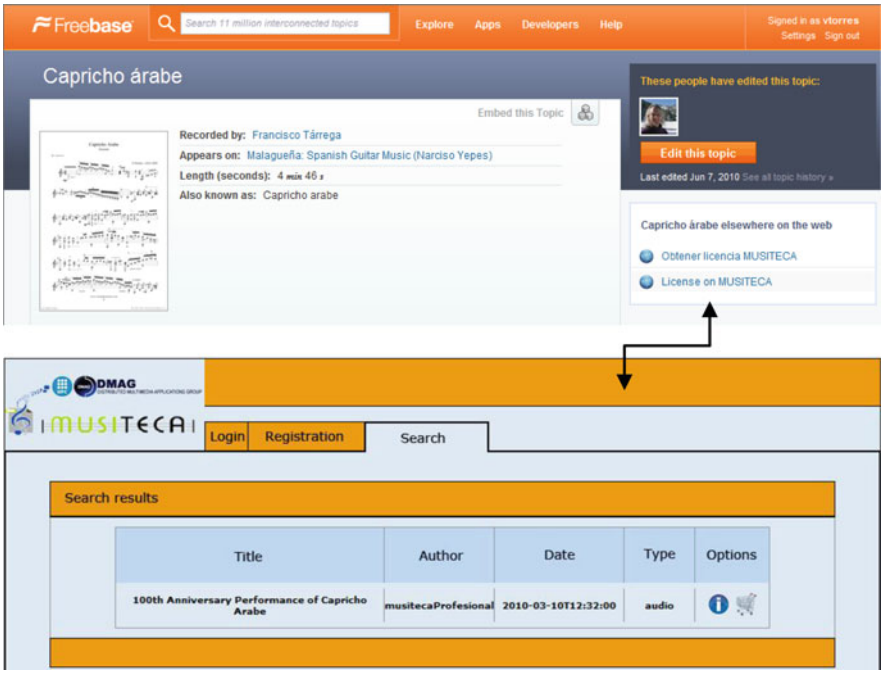


Fig. 25.3 Licensing link from Freebase to a specific trading portal in Musiteca

NetPortedItems [32], a Spanish SME company. This CMS provides content authors the possibility of registering their work into the CMS. An author may even describe how other authors can use their work for deriving new content. This information is described using LS licenses, where we have added a special condition called Rights over Derivatives (ROD) [33, 34]. This condition indicates the percentage of the income that someone gets from a derivative work that is required to return to the original author. When an author creates derived content from an existing work and gets any revenue from it, the CMS follows back the chain of works, calculates the money for each author from the ROD condition and creates a report for each author informing of this fact. Reports can be consulted at established time periods to give each author the corresponding revenues. This system makes use of all MIPAMS services through a dedicated portal. Figure 25.4 shows a sample screenshot.

25.4.4 Content Registration, Search, Licensing and Access Control without Content Management

This scenario has been implemented in CulturaLive [35], a research project funded by the Catalan Administration. In this project, we have integrated, using Web

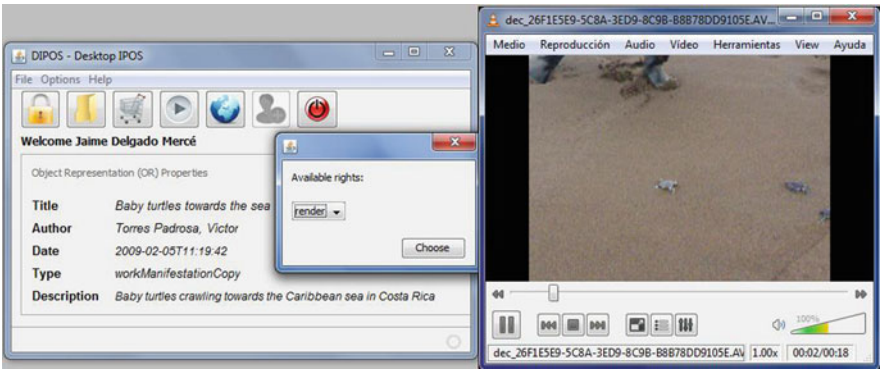


Fig. 25.4 Content access and unprotected rendering in IPOS

Services, MIPAMS LS, AS and RS into an existing system offered by another project partner [36] that provides audiovisual content to be broadcasted live through Digital Terrestrial Television (DTT) by televisions participating in the project. With our modules, content purchases can be tracked, as we register each license acquisition and authorization result (positive or negative) into a reporting database. This database can be later consulted for billing purposes. It is worth noting that digital content to be broadcasted is not managed by MIPAMS but directly by the different TV channels and SME's in the project consortium. This gives an idea of the integration capabilities of the MIPAMS platform.

25.4.5 Content Registration, Protection, Search, Licensing and Access Control for Limited Capabilities Devices

This scenario has been implemented in some projects of our research group (e.g. AXMEDIS [25], but also in other projects) in a slightly different way. In such projects, the modules involved in the authorization of user actions were located inside the mobile device. In this way, when the user wanted to consume some content, the license for authorizing this action was inside the mobile. This was done to avoid calling external services, as it involved a phone call or data transaction that might involve a non-negligible cost for the user. Moreover, mobile devices used a specific licensing schema (OMA DRM [37]) addressed to devices with limited processing and communication capabilities. Currently, since smartphones and high capacity mobile devices are gaining relevance and current telecommunications companies are adopting competitive pricing policies for mobile users (e.g. flat fees), the solutions being implemented might be reconsidered.

To implement this scenario with MIPAMS, if content is already registered and protected using a protection mechanism non-compatible with the device, the intermediary could be responsible for decrypting content and reencrypting it to

Fig. 25.5 Link to external sites for licensing content

deal with the device limitations. Otherwise, if content is only to be used by limited capabilities devices, it should be encrypted using the suitable protection mechanism when uploaded to the CS.

25.4.6 Content Registration and Licensing Through External Services

Figure 25.5 shows how content could be linked from the MIPAMS-based intellectual property repository developed in the Musiteca [30] project towards external specialized licensing portals. Some examples (not used in this project) are YouLicense [38] or Getty Images [39]. Content would be registered and accessible for being searched, while the shopping chart icon would redirect the user to a specialized and external licensing service.

25.5 Conclusions

In this chapter, we have presented three initiatives that deal with the development of distributed multimedia applications and Digital Rights Management technologies. In this context, we have proposed several relevant usage scenarios that share some common functionality such as content registration, protection, search, licensing and access control.

We have also presented some sample implementations done by the DMAG in different research projects and we have proved how the aforementioned functionality can be integrated into a single generic architecture called MIPAMS, which offers distributed services and enables to build specific applications on top of it, that depend on the business model being followed.

We plan to continue contributing from our developments to the standards in progress and adopt new standards specifications when possible, to facilitate interoperability. Furthermore, we are progressing in the development and exploitation of the architecture and applications on top of it.

Acknowledgements Part of this work has been co-funded by the Spanish administration: Multimedia Content Management Life Cycle (MCM-LC) project (TEC2008-06692-C02-01) and Musiteca project (TSI-020501-2008-117); by the European Commission: AXMEDIS (IST-2004-511299); by the Catalan administration: CulturaLive; and by the company NetPortedItems, S.L.: IPOS.

References

1. ISO/IEC: MPEG extensible middleware. http://mxm.wg11.sc29.org (2010)
2. ISO/IEC: ISO/IEC JTC1 SC29/WG11 N11230, context and objectives for advanced IPTV terminal (AIT), Kyoto, Japan (2010)
3. Distributed Multimedia Applications Group (DMAG): http://dmag.ac.upc.edu (2010)
4. Torres, V., Delgado, J., Llorente, S.: An Implementation of a Trusted and Secure DRM Architecture. Lecture Notes in Computer Science 4277. Springer, Heidelberg (2006)
5. Delgado, J., Rodríguez, E.: Digital rights management technologies and standards. In: Ng, K., Nesi, P. (eds.) Interactive Multimedia Music Technologies. Information Science Reference, New York (2008)
6. Creative Commons Licenses: http://creativecommons.org/licenses/ (2010)
7. ISO/IEC: ISO/IEC IS 21000-2: information technology—multimedia framework (MPEG-21)—part 2: digital item declaration (2005)
8. ISO/IEC: ISO/IEC IS 21000-5: information technology—multimedia framework (MPEG-21)—part 5: rights expression language (2004)
9. ISO/IEC: ISO/IEC IS 21000-4: information technology—multimedia framework (MPEG-21)—part 4: intellectual property management and protection components (2006)
10. ISO/IEC: ISO/IEC IS 21000-15: information technology—multimedia framework (MPEG-21)—part 15: event reporting (2006)
11. OASIS: Security assertion markup language (SAML). http://saml.xml.org/ (2005)
12. Rodríguez, V., Delgado, J., Chiariglione, F., et al.: Interoperable Digital Rights Management Based on the MPEG Extensible Middleware, Multimedia Tools and Applications. Springer, Dordrecht (2010)
13. ISO/IEC: ISO/IEC 23006-1: information technology—MPEG-M (MPEG extensible middleware)—part 1: MXM architecture and technologies. Final Draft International Standard (2010)
14. ISO/IEC: ISO/IEC 23006-2: information technology—MPEG-M (MPEG extensible middleware)—part 2: MXM API. Final Draft International Standard (2010)
15. ISO/IECL: ISO/IEC 23006-3: information technology—MPEG-M (MPEG extensible middleware)—part 3: MXM reference software. Final Draft International Standard (2010)
16. ISO/IEC: ISO/IEC 23006-4: information technology—MPEG-M (MPEG extensible middleware)—part 4: MXM protocols. Final Draft International Standard (2010)
17. ISO/IEC: ISO/IEC IS 21000-7: information technology—multimedia framework (MPEG-21)—part 7: digital item adaptation (2007)
18. Llorente, S., Delgado, J., Maroñas, X.: Implementing mobile DRM with MPEG-21 and OMA. In: Proceedings of the 5th International Workshop on Security in Information Systems, pp. 166–175 (2007)

19. ISO/IEC: ISO/IEC IS 21000-19: information technology—multimedia framework (MPEG-21)—part 19: media value chain ontology (2010)
20. ISO/IEC: ISO/IEC JTC1 SC29/WG11 N11228, requirements for advanced IPTV terminal (AIT), Kyoto, Japan (2010)
21. ISO/IEC: ISO/IEC JTC1 SC29/WG11 N11336, advanced IPTV terminal (AIT): 2nd call for proposals, Dresden, Germany (2010)
22. ISO/IEC: ISO/IEC JTC1 SC29/WG11 N11435, evaluation report on AIT contributions, Geneva, Switzerland (2010)
23. ISO/IEC. In: Chiariglione, F., Gauvin, M., Huang, T., Rodríguez, V., Delgado, J., Grafl, M., Matone, S. (eds.): ISO/IEC JTC1 SC29/WG11 N11413, working draft of ISO/IEC 23006-4 basic services, Geneva, Switzerland (2010)
24. ISO/IEC. In:Gallo, F., Grafl, M., Matone, S. (eds.): ISO/IEC JTC1 SC29/WG11 N11414, working draft of ISO/IEC 23006-5 service aggregation, Geneva, Switzerland (2010)
25. AXMEDIS (IST-2004-511299): Automating production of cross media content for multi-channel distribution. http://www.axmedis.org, European Commission (2004–2008)
26. ISO/IEC. Rodríguez, V., Delgado, J., Rodríguez, E., Llorente, S. (eds.): ISO/IEC JTC1 SC29/WG11 M17561, DMAG-UPC response to the AIT call, Dresden, Germany (2010)
27. ISO/IEC. Delgado, J., Rodríguez, V., Rodríguez, E. (eds.): ISO/IEC JTC1 SC29/WG11M18561, proposal of initial contract expression language working draft, Guangzhou, China (2010)
28. Intellectual Property Operations System (IPOS): http://dmag1.ac.upc.edu/IPOS (2010)
29. Torres, V., Rodríguez, E., Llorente, S., Delgado, J.: Architecture and Protocols for the Protection and Management of Multimedia Information. Lecture Notes in Computer Science 3311. Springer, Heidelberg (2004)
30. Musiteca Research Project (TSI-020501-2008-117): Ministerio de Industria, Turismo y Comercio (Subprograma Avanza I + D) (2008)
31. Musiteca Freebase Database: http://musiteca.freebase.com/ (2010)
32. NetPortedItems, S.L.: http://www.digitalmediavalues.com/ (2010)
33. Torres, V., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M.: A web-based rights management system for developing trusted value networks. In: Proceedings of the 18th International World Wide Web Conference Developer's Track, pp. 57–59 (2009)
34. Torres, V., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M.: Enhancing rights management systems through the development of trusted value networks. In: Proceedings of the 7th International Workshop on Security in Information Systems, pp. 26–35 (2009)
35. CulturaLive research Project (2009REGIÓ 00024): Generalitat de Catalunya (2009)
36. Video Stream Networks (VSN): http://www.vsn-tv.com/es (2010)
37. Open Mobile Alliance Digital Rights Management (OMA DRM): http://www.openmobilealliance.org/technical/release_program/drm_v2_1.aspx (2010)
38. YouLicense: http://www.youlicense.com/ (2010)
39. Getty Images: http://www.gettyimages.com/ (2010)

Chapter 26
Scalable Video Coding in Content-Aware Networks: Research Challenges and Open Issues

Michael Grafl, Christian Timmerer, Hermann Hellwagner, Daniel Negru, Eugen Borcoci, Daniele Renzi, Anne-Lore Mevel and Alex Chernilov

Abstract The demand for access to advanced, distributed media resources is nowadays omnipresent due to the availability of Internet connectivity almost anywhere and anytime, and of a variety of different devices. This calls for rethinking of the current Internet architecture by making the network aware of which content is actually transported. This paper introduces Scalable Video Coding (SVC) as a tool for Content-Aware Networks (CANs) which is currently researched as part of the EU FP7 ALICANTE project. The architecture of ALICANTE with respect to SVC and CAN is presented, use cases are described, and finally research challenges and open issues are discussed.

Keywords Content-aware networking · Network-aware applications · Scalable video coding · Quality of service · Multimedia distribution · Future Internet

M. Grafl (✉) · C. Timmerer · H. Hellwagner
Klagenfurt University, Universitätsstraße 65–67, 9020 Klagenfurt, Austria
e-mail: michael.grafl@itec.uni-klu.ac.at

C. Timmerer
e-mail: christian.timmerer@itec.uni-klu.ac.at

H. Hellwagner
e-mail: hermann.hellwagner@itec.uni-klu.ac.at

D. Negru
e-mail: daniel.negru@labri.fr

E. Borcoci
e-mail: Eugen.Borcoci@elcom.pub.ro

D. Renzi
e-mail: daniele@bsoft.net

A.-L. Mevel
e-mail: Annelore.Mevel@grassvalley.com

A. Chernilov
e-mail: Alexc@optibase.com

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_26,

26.1 Introduction

In recent years the number of contents, devices, users, and means to communicate over the Internet has grown rapidly and with that the heterogeneity of all the involved entities. Many issues can be associated with that which are generally referred to as ongoing research in the area of the Future Internet (FI) [1]. One project in this area is the European FP7 Integrated Project "Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments" (ALICANTE) [2] which proposes a novel concept towards the deployment of a new networked *Media Ecosystem*. The proposed solution is based on a flexible cooperation between providers, operators, and end users, finally enabling every user (1) to access the offered multimedia services in various contexts, and (2) to share and deliver her/his own audiovisual content dynamically, seamlessly, and transparently to other users.

Towards this goal, ALICANTE's advanced concept provides *content-awareness* to the network environment, *context-awareness* (network/user) to the service environment, and *adapted services/content* to the end user for her/his best service experience possible, where the end user can take the role of a consumer and/or producer. The term *environment* denotes a grouping of functions defined around the same functional goal and possibly spanning, vertically, one or more architectural (sub-)layers. This term is used to characterize a broader scope than the term *layer*.

The ALICANTE architecture introduces two novel virtual layers on top of the traditional network layer, i.e., a Content-Aware Network (CAN) layer for packet processing at network layer and a Home-Box (HB) layer for the actual content adaptation and delivery. Furthermore, Scalable Video Coding (SVC) is heavily employed for the efficient, bandwidth-saving delivery of media resources across heterogeneous environments (cf. Sect. 26.2). Technical use cases that will benefit from this architecture are outlined in Sect. 26.3, and Sect. 26.4 details the research challenges and open issues to be addressed in the course of the project. Finally, the chapter is concluded in Sect. 26.5.

26.2 ALICANTE: Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments

26.2.1 Overview and System Architecture

The ALICANTE architecture promotes advanced concepts such as content-awareness to the network environment, user context-awareness to the service environment, and adapted services/content to the end user for her/his best service experience, for both consumer and producer roles.

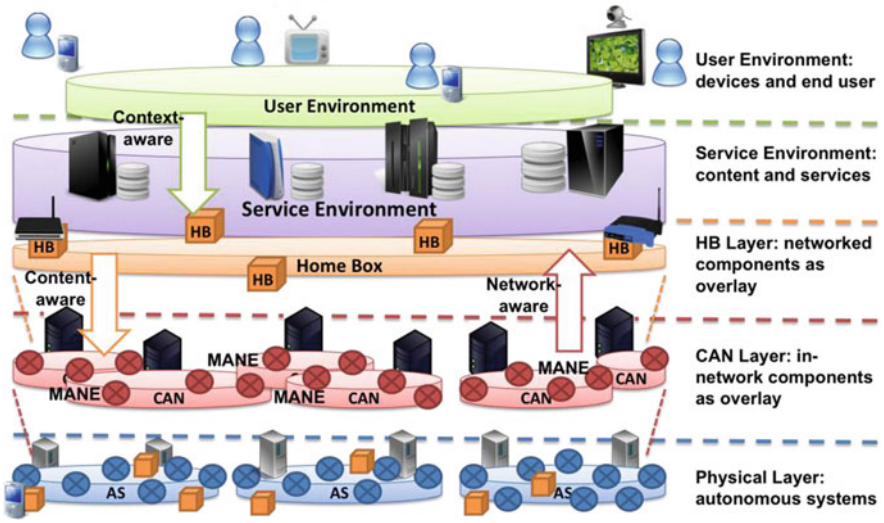


Fig. 26.1 ALICANTE concept and system architecture

Two novel virtual layers are proposed on top of the traditional network layer as depicted in Fig. 26.1: the *Content-Aware Network* (*CAN*) *layer* for network packet processing and a *Home-Box* (*HB*) *layer* for the actual content adaptation and delivery.

Innovative components instantiating the CAN are called *Media-Aware Network Elements* (*MANE*). They are actually CAN-enabled routers and associated managers, offering together content-aware and context-aware Quality of Service/Experience (QoS/QoE), content-aware security, and monitoring features, in cooperation with the other elements of the ecosystem.

The upper layer, i.e., the *Service Environment*, uses information delivered by the CAN layer and enforces network-aware application procedures, in addition to user context-aware ones. The Service Environment comprises Service Providers and Content Providers (SP/CP) which offer high-level media services (e.g., video streaming, video on demand, live TV) to the end users.

The novel proposed *Home-Box* (*HB*) entity is a physical and logical entity located at end users' premises which is gathering context, content, and network information essential for realizing the big picture. Associated with the architecture there exists an open, metadata-driven, interoperable middleware for the adaptation of advanced, distributed media resources to the users' preferences and heterogeneous contexts enabling an improved Quality of Experience. The adaptation will be deployed at both the HB and CAN layers making use of scalable media resources as outlined in the next section.

For more detailed information the interested reader is referred to [3].

26.2.2 Scalable Video Coding and Content-Aware Networks

The adaptation relies on Scalable Video Coding (SVC) [4]. SVC follows a layered coding scheme comprising a base layer and one or more enhancement layers with various dimensions. Three basic scalable coding modes are supported, namely spatial scalability, temporal scalability, and Signal to Noise Ratio (SNR) scalability, which can be combined into a single coded bit stream:

- *Spatial (picture size) scalability*. A video is encoded at multiple spatial resolutions. By exploiting the correlation between different representations of the same content with different spatial resolutions, the data and decoded samples of lower resolutions can be used to predict data or samples of higher resolutions in order to reduce the bit rate to code the higher resolutions.
- *Temporal (frame rate) scalability*. The motion compensation dependencies are structured so that complete pictures (i.e., their associated packets) can be dropped from the bit stream. Note that temporal scalability is already enabled by the Advanced Video Coding (AVC) standard and that SVC only provides supplemental enhancement information to improve its usage.
- *SNR/Quality/Fidelity scalability*. Each spatial resolution is encoded at different qualities. The data and decoded samples of lower qualities can be used to predict data or samples of higher qualities in order to reduce the bit rate to code the higher qualities.

The adaptation deployed at the CAN layer will be performed in a Media-Aware Network Element (MANE) [5]. MANEs, which receive feedback messages about the terminal capabilities and delivery channel conditions, can remove the non-required parts from a scalable bit stream before forwarding it. Thus, the loss of important transmission units due to congestion can be avoided and the overall error resilience of the video transmission service can be substantially improved.

Design options of in-network adaptation of SVC have been described in previous work [6] and first measurements of SVC-based adaptation in an off-the-shelf WiFi router have been reported in [7]. More complex adaptation operations that will be required to create scalable media resources, such as transcoding [8] of media resources which have increased memory or CPU requirements, will be performed at the edge nodes only, i.e., in the Home-Boxes. Therefore, the ALICANTE project will develop an SVC (layered-multicast) tunnel, as depicted in Fig. 26.2, inspired by IPv6 over IPv4 tunnels. That is, within the CAN layer only scalable media resources—such as SVC—are delivered adopting a layered-multicast approach [9] which allows the adaptation of scalable media resources by the MANEs implementing the concept of distributed adaptation. At the border to the user, i.e., the Home-Box, adaptation modules are deployed enabling device-independent access to the SVC-encoded content by providing X-to-SVC and SVC-to-X transcoding/rewriting functions, where X = {MPEG-2, MPEG-4 Visual, MPEG-4 AVC, etc.}. An advantage of this approach is the

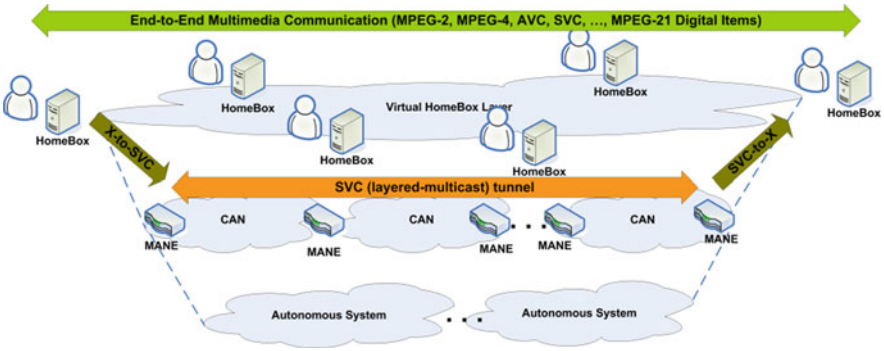


Fig. 26.2 Concept of SVC (layered-multicast) tunnel

reduction of the load on the network (i.e., no duplicates), making it free for (other) data (e.g., more enhancement layers). However, multiple adaptations may introduce challenges that have not been addressed in their full complexity (cf. Sect. 26.4).

The key innovations of the ALICANTE project with respect to service/content adaptation are as follows [10]:

- *SVC tunnels with in-network adaptation* will be set up, enabling *better network resource utilization* while maintaining a satisfactory QoE for the end user.
- Regarding the adaptation decision-taking framework, the project will aim at providing a *self-organizing control framework* including *adaptation decision aggregation* (for scalability reasons) and *propagation*.
- Means for *dynamic and intelligent adaptation of content at the Home-Box level* will be investigated, based on *distributed coordination with the CAN layer* to perform *optimal adaptation* and to *improve bandwidth* and *Home-Box's processing power usage*.
- A *metadata-driven in-network adaptation solution at the Content-Aware Network* level will be deployed enabling *dynamic adaptation based on context information* (terminal capabilities and network characteristics).
- Finally, a *test-bed and subjective tests for evaluating the QoE* for given use cases will provide the feedback for fine-tuning the parameters for the adaptation at the Home-Box and CAN levels.

26.3 Use Cases

In order to evaluate the concept of SVC in the context of CANs/HBs, several use cases have been defined, a selection thereof is briefly introduced in the subsequent sections.

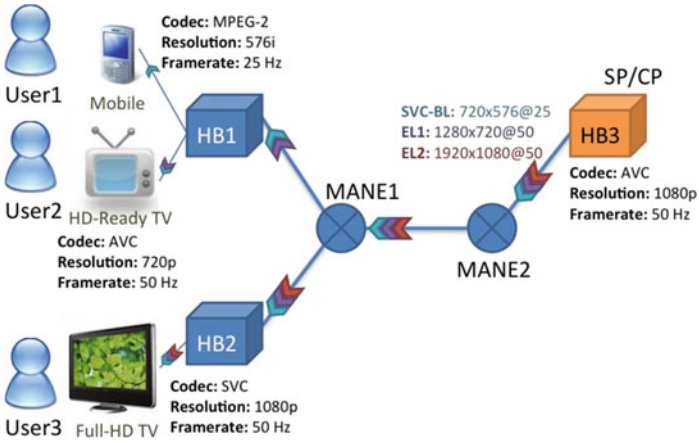


Fig. 26.3 Multicast/broadcast with SVC adaptation

26.3.1 Multicast/Broadcast

In this scenario, multiple users are consuming the same content from a single provider (e.g., live transmission of sport events). The users may have different terminals with certain capabilities as depicted in Fig. 26.3. The ALICANTE infrastructure is simplified in Fig. 26.3 to highlight the interesting parts for this scenario (i.e., the HBs and the MANEs). Note that the SVC layers depicted in the figure are only examples and that SVC streams in ALICANTE may comprise temporal, spatial, and quality (SNR) scalability with multiple layers. The properties and numbers of SVC layers will be determined by the HB at the Service/Content Provider (SP/CP) side based on several parameters (e.g., diversity of terminal types, expected network fluctuations, size overhead for additional layers, available resources for SVC encoding/transcoding, etc.) which are known a priori or dynamically collected through a monitoring system operating across all network layers.

26.3.2 Home-Box Sharing

In this scenario, a user consumes content through a foreign (shared) HB, e.g., the user accesses the content/service to which she/he has subscribed while being abroad (e.g., business trip, vacation). Figure 26.4 depicts a user consuming content at two different locations on two different terminals, connected to different HBs. Note that the user might as well use her/his mobile phone to consume content through HB2.

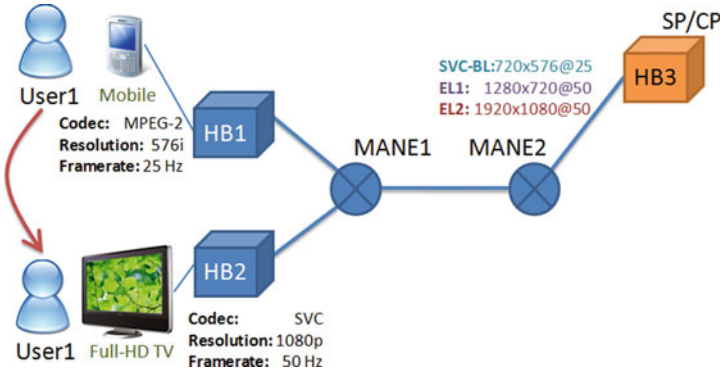


Fig. 26.4 Home-box sharing

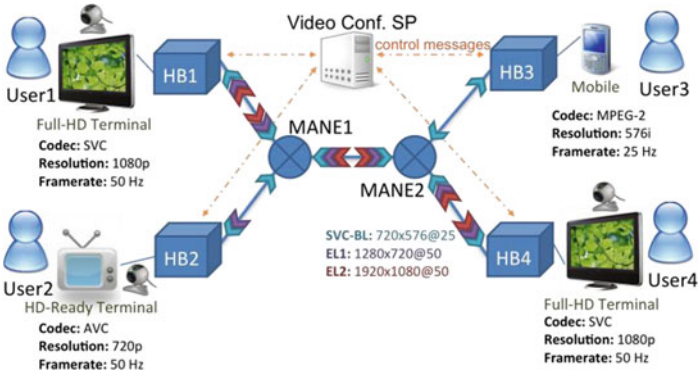


Fig. 26.5 Video conferencing

26.3.3 Video Conferencing

This scenario consists of an m:n video conferencing session (e.g., in family meetings, office meetings, etc.) as depicted in Fig. 26.5. The media distribution is handled over a multicast shared bi-directional non-homogeneous tree in the ALICANTE network. In such a way only the minimum amount of network resources are spent, while assuring maximum quality to the end user.

26.3.4 Peer-to-Peer Media Streaming

The HBs operate in peer-to-peer (P2P) mode within the ALICANTE ecosystem as illustrated in Fig. 26.6. The MANEs, through which the P2P traffic flows, act as proxy caches which intercept requests for content pieces issued by HBs and

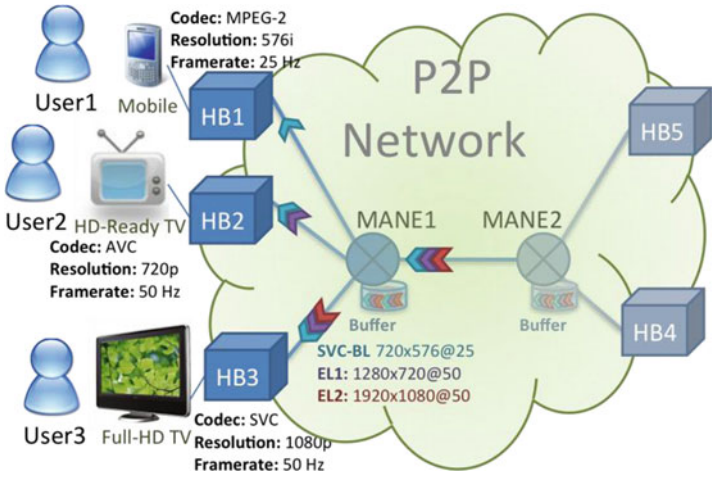


Fig. 26.6 P2P media streaming

aggregate them respecting the capabilities of requesting terminals. Furthermore, content pieces are only forwarded if the requesting terminals can decode them. Therefore, unnecessary traffic is reduced to a minimum freeing up the network resources for other data (e.g., additional enhancement layers).

26.4 Research Challenges and Open Issues

In this section we point out some research challenges and open issues with respect to utilizing Scalable Video Coding within Content-Aware Networks.

Distributed adaptation decision-taking framework. Due to the fact that many, possibly heterogeneous entities are involved—in the production, ingestion, distribution, and consumption stages—there is a need to develop a framework for distributed adaptation decision-taking; that is, finding the optimal decision regarding the adaptation of the content for a single entity (i.e., HB, MANE) within a network of various entities in the delivery system. Note that decision-taking is needed at the request stage and during the delivery of the multimedia content as (network) conditions might change.

Distributed adaptation at HB and CAN layers. The actual adaptation at both layers needs to be done efficiently, based on several criteria, in order to obtain low (end-to-end) delay, minimum quality degradation, and assuring scalability in terms of the number of sessions that can be handled in parallel.

Efficient, scalable SVC tunneling and signaling thereof. The approach of tunneling the content within SVC streams in the (core) network opens up a number of issues due to SVC adaptation within the MANEs, SVC transcoding/rewriting within the HBs, and the associated signaling requirements. The issues range from efficiency and scalability to quality degradations and latency.

Impact on the quality of service/experience (QoS/QoE). As there may be many adaptations happening during the delivery of the content, the impact on QoS and QoE needs to be studied in order to find the best trade-off for the use cases in question. While for the QoS many objective measures are available, the QoE is highly subjective and requires tests involving end users; these tests are time consuming and costly. In any case, a good test-bed is needed for both objective and subjective tests for the evaluation of the QoS and QoE, respectively. The possible mappings between QoS and QoE will be considered in this work also.

Cooperation between the adaptation framework and CAN overlay management. While the adaptation framework operates mainly at flow level, the CAN management deals with control information at an aggregated level. Appropriate cooperation between them and mappings for monitoring and control information have to be defined in order to ensure efficient use of transport resources.

26.5 Conclusions and Future Work

In this chapter we have introduced the usage of Scalable Video Coding in Content-Aware Networks for various use cases. In particular, SVC is a promising tool for making the network aware of the actual content being delivered, i.e., when it comes to technical properties such as bit rate, frame rate, and spatial resolution. Furthermore, it allows for efficient and easy-to-use in-network adaptation due to the inherent structure of SVC. The goal of the ALICANTE project is to provide an advanced Media Ecosystem that enables the management of media services with respect to QoS and QoE on the one hand, while delivering the media content at dynamically adaptable bit rates to heterogeneous terminals on the other hand.

The use cases described in this chapter indicate the advantages of using SVC and in-network adaptation and we have highlighted research challenges and open issues. However, as this work is in its early stage it lacks validation results for the scenarios and solutions proposed which remains part of our future work.

Acknowledgments This work is supported in part by the European Commission in the context of the ALICANTE project (FP7-ICT-248652) (http://www.ict-alicante.eu/).

References

1. Tselentis, G., et al.: Towards the Future Internet—emerging trends from European research. IOSPress, Amsterdam (2010)
2. ALICANTE web site. http://www.ict-alicante.eu/. Accessed 09 Oct 2010
3. Borcoci, E., Negru, D., Timmerer, C.: A novel architecture for multimedia distribution based on content-aware networking. In: Proceedings of the Third International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ2010), Athens/Glyfada, Greece (2010)

4. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst. Video Technol. **17**(9), 1103–1120 (2007)
5. Wenger, S., Wang, Y.-K., Schierl, T.: Transport and signaling of SVC in IP networks. IEEE Trans. Circuits Syst. Video Technol. **17**(9), 1164–1173 (2007)
6. Kuschnig, R., Kofler, I., Ransburg, M., Hellwagner, H.: Design options and comparison of in-network H.264/SVC adaptation. J. Vis. Commun. Image Represent. **19**(8), 529–542 (2008)
7. Kofler, I., Prangl, M., Kuschnig, R., Hellwagner, H.: An H.264/SVC-based adaptation proxy on a WiFi router. In: 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV2008), Braunschweig, Germany (2008)
8. Shen, B., Tan, W.-T., Huve, F.: Dynamic video transcoding in mobile environments. IEEE Multimed. **15**(1), 42–51 (2008)
9. McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven layered multicast. In: SIGCOMM (1996)
10. Grafl, M., Timmerer, C. (eds.): Service/Content Adaptation Definition and Specification, ICT ALICANTE, Deliverable D2.2 (2010)

Chapter 27
Network Layer Solutions for a Content-Centric Internet

Andrea Detti and Nicola Blefari-Melazzi

Abstract Nowadays most people exploit the Internet to get contents such as web pages, music or video files. These users only value "what" they download and are not interested about "where" content is actually stored. The IP layer does the opposite and cares about the "where" and not about the "what". This contrast between the actual usage of the Internet and the service offered by the IP layer is deemed to be the source of several problems concerning usability, performance, security and mobility issues. To overcome this contrast, research on the Future Internet is exploring novel so-called content-centric architectures, where the network layer directly provides users with contents, instead of providing communication channels between hosts. In this paper, we identify the main functionalities of a content-centric network (CONET), we discuss pros and cons of literature proposals for an innovative, content-centric network layer and we draw our conclusions, stating some general requirements that, in our opinion, a CONET should satisfy.

Keywords Content-distribution networks · Domain names · IP · Content routing

27.1 Introduction

There is a growing consensus in the recent literature that the central role of the IP address poorly fits the actual form of Internet usage. A typical user does not type IP addresses; she gets data or services by using application tools (e.g., Google,

A. Detti (✉) and N. Blefari-Melazzi
Electronic Engineering Department, University of Rome "Tor Vergata", Rome, Italy
e-mail: andrea.detti@uniroma2.it

N. Blefari-Melazzi
e-mail: blefari@uniroma2.it

N. Blefari-Melazzi et al. (eds.), *Trustworthy Internet*,
DOI: 10.1007/978-88-470-1818-1_27,

YouTube, Facebook, Skype), which operate on the basis of a description of the desired content. This means that users actually exploit the Internet in a content-centric way; indeed, they are not interested in knowing from "where" contents are provided, they are only interested in the fact that they can get "what" they want. Conversely, the underlying IP communication model is still address-centric (or host-centric); that is, the network layer has to be fed by IP addresses, which are used to ascertain from "where" contents have to be taken. Therefore, there is a mismatch between the content-centric usage model of the Internet and the address-centric service model offered by the IP layer. Such a mismatch gives rise to several problems that would not exist if the network layer were a content-centric one [1–3]. Some of these issues are:

- *Persistence of the Names*. Once a user gives a name to a content, she wishes that the name remains valid even if the provider that makes available the content on the Internet changes. Today, naming is based on the WEB URL structure, which has a "where/what" format. The URL structure is perfectly tailored for an address-centric network layer, but it implies the change of the name in case the provider (i.e., "where") changes. If the network-layer were a content-centric one, it would be able to route on the basis of the content (routing-by-name), thus avoiding the need of including "where" in the name of the content.
- *Content distribution*. Today the reliability of a content and the time required for retrieving a content are improved by distributing (caching or pre-fetching) replicas of the content on different servers, geographically distributed. Furthermore, content replication strongly limits the amount of traffic traversing the backbone, since a lot of data sessions are locally handled by these servers. Since the IP layer is unable to "understand" contents, content distribution is achieved by means of proprietary application-layer systems [4], like Akamay CDN, or WEB proxies. While the success of these systems is undeniable, they do not cooperate with each other, often they are not free of charge and they are not available everywhere; thus, their potential effectiveness is reduced. If the network layer were a content-centric one, it would be aware of which contents are traversing the network and could autonomously and natively implement replication strategies, everywhere and for all [5].
- *Lookup delay*. Today the real delivery of a content begins only after that the client application has queried the DNS server, in order to discover the IP address of the related server. The additional DNS request-response delay is quickly becoming a dominant delay factor [1]; indeed Internet data-rates are faster growing, reducing the time needed to transfer content. If the network layer were content-centric, it would route directly the request toward the content, thus avoiding the mediation of a DNS server.
- *Mobility*. An IP address identifies both the location and the identity of an end-point; this location-identity tie is deemed to be the source of many mobility-related weaknesses. A content-centric network overcomes these limitations by basing the routing directly on the identity of an end-point; in this case the end-

point is the actual content and its identity (e.g., a name) does not include any reference to its location.

- *Security*. Users are interested in receiving trusted contents; today this is achieved in an indirect way: a user trusts "who" provides the content, rather than trusting the content itself. This approach could be risky as there are a lot of actors to trust, in the process of content retrieval [6]. For instance, a user believes that she is trading with Amazon because she clicks on a link with the name "http://www.amazon.com", which is provided by a search-engine tool, like Google. In doing so, she is trusting both the search-engine and the DNS server providing the name-to-address translation. Moreover, she is also trusting the mirror server where she could be redirected by a content distribution system [4]. The higher the number of actors to trust, the more critical is the overall security of the system. In a content-centric network, security information travels with the content, so that even if content is provided by an un-trusted server or is transferred on an un-trusted network, it can be validated by the consumer. Moreover, such a content-based security also enables the replication of secure contents by any network node, which is very important as users can get contents not only from the original content creator or source node but also from any node/user that has already downloaded that content.

Currently, several researchers and research projects are proposing their network architectures and protocol stacks aimed at implementing the content-centric paradigm. In terms of deployment, the proposed solutions can be classified as evolutionary [1, 2, 7] or clean-slate [3, 8–10]. The evolutionary architectures implement content-centric functionalities by enhancing the actual Internet. The enhancement consists in deploying new entities that form a content-centric "overlay" network. Conversely, clean-slate architectures are alternative to the TCP/IP one. A clean-slate architecture is devised to operate directly over the link-layer and, therefore, implies a redesign of network layer functionalities from scratch. Obviously, a clean-slate solution can also be implemented at the overlay level, e.g., by visualizing IP tunnels as link-layer interface; however, in this case some functionalities could be duplicated in TCP/IP and content-centric layers.

In this paper, we outline the principles of a content-centric network layer and discuss pros and cons of literature approaches. Of course, a complete system requires also functionalities provided by higher protocol layers, in addition to the network ones; however, as of today, the main research effort focuses on the network layer, considered as the main pillar of a future content-centric Internet.

27.2 Principles of a Content-Centric Network Layer

In this section we discuss the main principles of a content-centric network layer, i.e. we deal with the protocol functionalities that should be implemented in each router of a clean-slate content-centric network architecture. A content-centric network layer should:

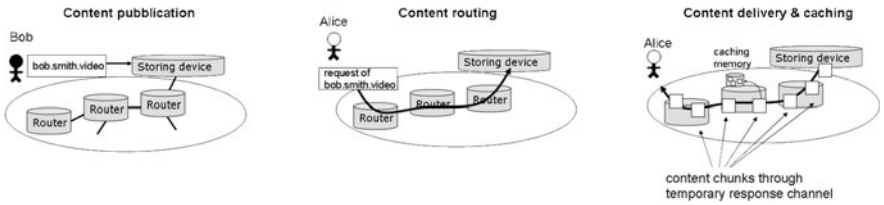


Fig. 27.1 Use-case

- Address contents, rather than hosts, adopting an addressing scheme based on names, which do not include any reference to their location.
- Route a user request, which includes a "destination" content-name, toward the closest copy of the content with such a name (name-based anycast routing).
- Deliver contents to address-less hosts.
- Provide a native, in-network caching to achieve efficient content delivery both in fixed and mobile environments.
- Exploit security information embedded in the content so as to avoid the diffusion of fake versions of contents.

For the sake of clarity we describe all these functionalities in a use-case, depicted in Fig. 27.1. Bob Smith is a young reporter that has recorded a video and wishes to publish it on a content-centric network. Bob prepares a data-package that embeds video and security information, he names this package as "bob.smith.video" and puts it in a storing device, connected with one or more network routers. Such a data-package is the actual content. Alice wishes to retrieve Bob's video and submits a request to the network layer, where the "destination" content-name is bob.smith.video. Each network router forwards Alice's request toward the closest storing device that holds a copy of Bob's content (exploiting only the name "bob.smith.video"); we name this phase "content-routing". When Alice's request reaches a storing device, the network layer delivers Bob's content to the address-less device of Alice; we name this phase "content-delivery". During the content-delivery phase, some intermediate routers may decide to cache and then forward [5] the content. When the content is received by Alice's device, the whole security information is checked to verify, for instance, the authenticity of the embedded video, the right of Alice to play it and so forth.

This scenario involves a set of functionalities like naming, routing, delivery and caching, which we discuss in the next subsections by presenting some recent literature proposals.

27.2.1 Content-Naming

In a content-centric network, routing is based on the name of the content; therefore, choosing the structure of names is an important aspect of the overall architecture.

From the user point of view it is better not to have any protocol restriction, so as to freely name contents. From the routing protocol point of view, this translates in flat-labels [11], i.e. names are a sequence of unstructured bytes.

Zooko's Triangle [12] identifies possible tradeoffs among three desirable properties of a naming scheme of a network protocol. Zooko's Triangle states that names can enjoy simultaneously at most two of the following characteristics:

- *Secure*. A name surely addresses only a given content; i.e., names can not be forged for addressing fake contents.
- *Memorable*. A human can remember a name for a couple of hours since the first time she has seen it on the side of a moving bus (moving bus test).
- *Decentralized*. A name can be chosen in a distributed way, i.e. without the need of a centralized naming authority.

For instance, actual DNS names are secure (if we trust the DS system) and memorable. Nick-names used by proprietary applications (e.g., Skype) are memorable and decentralized. So-called *self-certifying names* [2, 13] are secure and decentralized.

As regards the latter solution it is worth recalling that a self-certified name is cryptographically constructed so that one can securely verify if the associated content is the original one. For instance, in [2] the authors propose that a publisher P of a content may autonomously label the content as L and the resulting self-certified name is the Digest (Pubkey$_p$):L, where Digest (Pubkey$_p$) is a digest of the public key used by the publisher to sign the data. If that the publisher ensures the uniqueness of label L among her published contents, a self-certifying naming schemes is secure and decentralized.

We argue that the security of names is a necessary property for a content-centric network. In the actual Internet, we can be (almost) sure to obtain an original content, e.g. the CNN homepage, because we trusts the DNS system, which provides the IP address of the original WEB server (or of an authoritative one). In a content-centric network, the retrieval of the content from the original server may be a rare event. Indeed, any *untrusted* router or device may make available its cached copy of the CNN homepage, and the network layer provides the user with the closest data named "CNN homepage". Therefore, it is fundamental that all contents named "CNN homepage" are equal to the original one; i.e., the name "CNN homepage" must not be forge-able (security property).

Since security is a must, given the Zooko's Triangle, we have to choose between decentralization and memorability. We think that this choice depends on requirements of applications, therefore we could have both secure-memorable and secure-decentralized [2] content-names. The memorability property is desirable for contents that need to be advertised, like the home-page of a business company or of a newspaper. Conversely, memorability may be not desirable for contents that have a very narrow interest, like a content describing the status of a DHL shipment, or the value measured by a sensor. Finally, we point out that implementing a secure-memorable naming for a content-centric network is currently an open issue (the DNS approach is not applicable in a content-centric network that

directly routes contents by name); also an open issue is how to allow the coexistence of secure-decentralized and secure-memorable naming schemes in the same network.

27.2.2 Content-Routing

The aim of content-routing is to forward a content request issued by an host toward a storage device that can provide such a content. With this regard, recent research proposals on future Internet [1–3, 8, 11] focus on devising a network layer anycast routing, which is "simply" based on names of contents, rather than on locations (i.e., IP address); consequently, in this paper, the term content-routing is synonymous with anycast routing-by-name. We observe that, before this wave of renewed interest, content-based routing has been widely studied for P2P overlay networks (e.g., [14]), improving also the simple routing-by-name with semantic functionality. Nevertheless, the inclusion of semantic functionality strongly increases the complexity of the architecture and may make critical its scalability at the Internet level.

Currently, the literature proposes two kinds of architectures, which implement the content-routing paradigm; we name these alternatives *advertise-based* and *rendezvous-based* architectures. In case of advertise-based architecture [1–3], routing protocols are practically the same of the actual IP ones (e.g., BGP), with the only difference that instead of advertising IP subnets, routers advertise the names of the contents that they can provide. Routing tables do not contain anymore IP addresses but content-names. Forwarding is executed by finding the best match between the name of the requested content and the entries of the routing tables.

A good example of a rendezvous-based architecture is the one proposed in [15]. The name of a content (and eventually other information, like the information-scope [8]), identifies a rendezvous-node (i.e., a location) by using a standardized function; all requests of that content will be forwarded towards that node by the network layer. The content provider stores in the rendezvous-node routing information, enabling the forwarding from the rendezvous-node to the node where content is effectively stored.[1] Therefore, the end-to-end path is composed of two parts: user-to-rendezvous and rendezvous-to-content. Routing protocols are inspired by overlay routing algorithms used in Distributed Hash Tables (DHTs) but they do not rely on an underlying network routing protocol [11, 16]

We argue that advertise-based architectures achieve optimal routing in terms of path length and are more effective in supporting content replication or caching, with respect to rendezvous-based ones; in fact, the same shortest-path anycast

[1] Obviously, if the content is uploaded directly on the rendezvous-node, this routing information is not necessary.

routing algorithm used for IP networks can be adopted for advertise-based architectures. In case of rendezvous-based architecture an end-to-end path is at the best stretched in two shortest-paths, and the absence of advertisements issued by nodes storing the same content (e.g., due to caching) makes more difficult to exploit content replication; i.e., the achievement of a perfect anycasting (see Sect. 27.2.4)

Advertise-based architectures seem to better use transport resources; however we argue that advertise-based architectures may suffer of routing scalability and stability issues. Scalability concerns arise if we observe that the number of entries of a routing table may be proportional to the number of content-names, unless an efficient and effective mechanism to compress name-based routing-tables will be devised. Nowadays IP routing is showing its scalability limits in terms of size of the routing tables [17, 18]; if we replace the actual IP prefix-based routing tables with name-based routing tables, surely the problem worsen as there are more contents than Autonomous Systems [1]. Moreover, we observe that the actual convergence delay of IP routing tables is short enough to cope with the actual dynamics of link creation and removal, thus ensuring the stability of the whole routing system. Conversely, the convergence delay of a name-based routing system has to cope with the dynamics of creation and removal of contents, and the occurrence of these events is more frequent than link creation and removal. Therefore, it is more difficult to ensure routing stability.

Finally, we observe that advertise-based architectures perfectly support a request-response interaction model but are less suitable for a publish-subscribe model [23]; indeed, it is difficult to handle the case of subscriptions issued before the corresponding publication, since routing tables are not configured in absence of actual content, i.e. of the publication. Conversely, in case of rendezvous-based architectures, a subscription can be routed and temporarily stored on the rendezvous-node, until the related publication gets available. However, rendezvous-based architectures can not easily control where in the network a rendezvous node handling a given content is located, e.g. in which geographical or administrative domain; a user publishing a content may not like that the related rendezvous node is located in the administrative domain of a competitor or in another country.

27.2.3 Content-Delivery

While content-routing is concerned with the host-to-content routing of user requests, content-delivery regards the content-to-host data transfer; i.e. the transferring of a content from its storage device to the requesting host.

In a content-centric network, hosts are not addressed by routing tables; i.e., routing tables contain only information about contents and not about hosts. This implies that, differently from IP, host-to-content and content-to-host routing involve different protocol functionalities.

Currently, the literature [3, 11, 19] copes with content-to-host routing by defining a temporary "response" channel, e.g. the sequence of downstream link-layer interfaces that content, or part of it, must follow from the storage device to the host. There are two proposed approaches to maintain the interface sequence: (i) the sequence may be explicitly contained in the header of the data-units transporting the content [19] (similarly to a source-routing approach); (ii) each downstream node may temporarily memorize information about the next-hop link-layer interface [3, 11] (similarly to the soft-state approach of the RSVP protocol).

In addition to the content-to-host routing issue, the delivery protocol may also include congestion control mechanisms. With this regard, the authors of [3] propose to download a content through a sequence of request-response interactions. At each interaction, a request conveys the interest for a number of "chunks" of the content; chunks have a fixed size (e.g., 512 kB) and the number of required chunks follows the dynamic of the TCP congestion control algorithm; nevertheless, differently from TCP, these chunks have to be network layer data-units, to allow caching (see Sect. 27.2.4)

27.2.4 Content-Distribution

Content-distribution concerns the possibility of caching (or replicating) a content or part of it in order to reduce the content-to-host path length. Reduction of path-length implies both a lower use of transport resource and, likely, a lower latency. In response to this challenge and considering the rapidly decreasing cost of memory, an approach that is attaining an increasing consensus is the "in-network" caching; i.e., to supply network routers with memory, to cache traversing contents [3, 5, 20].

Practically, to enable in-network caching, content delivery should be structured in a sequence of network data-units, which are stored, recognized by content-name and by a sequence-number, eventually cached, and forwarded hop-by-hop by each router on the content-to-host path. Obviously, a greater size of the data-units reduces the computation load of the caching operation, since routers would have a lower number of content parts to handle. Practically, it seems reasonable to have data-unit of 256–512 kB; therefore network layer data-units of a content-centric network should support sizes similar to the chunks of P2P file transfer applications (e.g., BitTorrent), rather than the typical sizes of IP packets. Obviously, increasing the size of the network data-units increases the store-and-forward delay; nevertheless the increasing of the line-speed should make negligible this kind of delay, if compared to the queueing one.

In-network caching could be carried out by routers either in an autonomous way or in a coordinated way [5]. In the autonomous way, a router could choose to cache a chunk of the content (i.e. a network data-unit) on the basis of a local algorithm, for instance based on the analysis of the request frequencies. In this case, nevertheless, closer routers have an high probability to store the same data, thus using

un-efficiently the whole system memory. Coordinated caching copes with this last issue by adopting caching algorithms based also on which are the data cached on closer routers.

Finally, in-network caching should inter-operate with content-routing in order to obtain an effective anycasting. This is simply achievable in case of advertise-based architecture, since every time a router has cached a content, the router advertises it. Conversely, in case of rendezvous-based architecture, the routing algorithm forwards user requests toward a specific rendezvous-node and then to a specific node storing the content; thus, if an intermediate router of the host-rendezvous-content path has a cached copy of the content, that router will serve directly the content request; nevertheless, it is more difficult to exploit also the caches of routers that are not on the path but that are close to the path (or elsewhere located).

27.3 Conclusions

A content-centric network is devised to directly provide what users value in the actual Internet, i.e. contents. This may imply a clean-slate re-design of the network layer, changing the actual communication paradigm (which has two hosts as end-points) in a novel paradigm (which has an host and a content as end-points). In this paper, we have classified and discussed pros and cons of some recent literature proposals coping with the main functionalities that a content-centric network layer should provide: content-naming, content-routing, content-delivery and content-distribution.

Although a content-centric architecture may better satisfy user needs, it is not easy to imagine a replacement of the actual TCP/IP infrastructure in the medium term. Thus, we argue that a content-centric network could more likely be deployed in a Future Internet consisting of software-routers running different network layer protocols, thus forming different virtual networks [21, 22].

Finally, we conclude the paper by stating some general requirements that, in our opinion, a content-centric network (CONET) should satisfy:

- A CONET should allow controlling where in the network contents or links to contents will be stored, e.g. in a given geographical or administrative domain; it is not acceptable that contents, or links to contents, are stored in "random" nodes, as it happens for instance in some solutions based on DHTs.
- A CONET should allow advertising the publication of a content within a limited geographical, application or administrative scope, e.g. for instance limiting such advertising to a definite section of the network (a campus, a shopping mall, an airport).
- A CONET should support not only persistent naming of specific pieces of content (e.g., a song, a CV, a movie) but should also support naming of a source of information with a consistent purpose (e.g. a meteo service) and of a source

of information made of possibly changing contents accessible with the same name (e.g. different edition of a news magazine).

- A CONET should support mechanisms to delete/update contents, supporting digital forgetting and garbage collection operations (e.g. giving to contents an expiry date and deleting all contents after the expiry date, or allowing users to delete/update contents that they generated and that are stored in the network and that they do not want anymore to be available to other users in their actual forms or at all).
- A CONET should allow selecting and retrieving the latest version (or earlier, or specific versions) of a series of contents identified by the same name (e.g. referenced by the same source of information).
- A CONET should allow supporting service sessions that require interactive exchange of data between two upper layers entities (e.g. a client server couple). This means that the network should provide functionality to deliver not only named content but also "un-named" content made up of data that are necessary to upper layer entities. Un-named contents can be any kind of data that upper layers need to exchange but that do not need to be named and do not need to be made accessible and identified in the network, per se. This is a key requirement necessary to natively support "traditional" client/server service (e.g. HTTP, POP, SMTP). For example, in the case of a client-server service session, un-named-contents are upper layer data (e.g. HTTP, SMTP, POP, SQL data) that are exchanged between a local client application and a remote server one (e.g. an HTTP server). Thanks to this functionality, a CoNet could natively support most of the actual Internet services, and any service that requires a point-to-point bidirectional interaction. We point out that this functionality extends the capabilities of a "plain" content-centric network and makes a CoNet suitable not only for content retrieval but for also for more traditional service deployment.
- A CONET should natively support a caching functionality without resorting to "external" mechanisms such as Content Distribution Networks. Caching shall be supported in principle by each node and also by user terminals. Users should be able to get the desired content from whatever node/terminal (e.g. the closest) and not necessarily from the original source, and even when disconnected from the CONET, but connected with a single other node/terminal that has a copy of the desired content. This requirement is fundamental to support recently proposed schemes such as wireless caching, distributed storage, recommendation strategies; such caching functionality, and more in general, making the network aware of the contents that is handling allows moving away the traffic from congested regions and balancing the load, for instance offloading cellular networks, or implementing other distribution strategies beneficial to network operators. This functionality would also give more control to network operators to handle traffic generated by so-called over-the-top players (such as Facebook, Youtube, etc), which generate vast amounts of traffic, leaving very little margin of actions to network operators.

Acknowledgements This work has been performed in the framework of the European-funded project CONVERGENCE (http://www.ict-convergence.eu). The project has received research funding from the Community's Seventh Framework programme.

References

1. Cheriton, D., Gritter, M.: TRIAD: a scalable deployable NAT-based internet architecture. Technical Report (2000)
2. Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. In: Proceedings of ACM SIGCOMM07 (2007)
3. Jacobson, V., Smetters, D.K., et al.: Networking named content. Fifth ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT) (2009)
4. Saroiu, S., Gummadi, K.P., Dunn, R.J., Gribble, S.D., Levy, H.M.: An analysis of Internet content delivery systems. SIGOPS Oper. Syst. Rev. **36**, 315–327 (2002)
5. Dong, L., Liu, H., Zhang, Y., Paul, S., Raychudhuri, D.: On the cache-and-forward network archiecture. In: Proceedings of the IEEE International Conference on Communications (ICC 2009) (2009)
6. Smetters, D.K., Jacobson, V.: Securing network content PARC, Technical Report, 2009 October
7. 4WARD, EU FP7 project. http://www.4ward-project.eu/
8. Publish-subscribe internet routing paradigm. EU FP7 Project. http://www.psirp.org/
9. Roberts, J.: The clean-slate approach to the future Internet design: a survey of research initiatives. Ann. Telecommun. **64**, 271–276 (2009)
10. Convergence, EU FP7 Project. http://www.ict-convergence.eu/
11. Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K., Stoica, I., Shenker, S.: ROFL: routing on flat labels. In: Proceedings of ACM SIGCOMM06 (2006)
12. Steigler, M.: An introduction to petname systems. http://www.skyhunter.com/marcs/petnames/IntroPetNames.html
13. Popescu, B.C., van Steen, M., Crispo, B., Tanenbaum, A.S., Sacha, J., Kuz, I.: Securely replicated web documents. In: Proceedings of IPDPS 2005
14. Carzaniga, A., Rutherford, M.J., Wolf, A.L.: A routing scheme for content-based networking. In: Proceedings of IEEE INFOCOM'04 (2004)
15. Stoica, I., Adkins, D., Zhuang, S., Shenker, S., Surana, S.: Internet indirection infrastructure. ACM SIGCOMM'02 (2002)
16. Caesar, M., Castro, M., Nightingale, E.B., O'Shea, G., Rowstron, A.: Virtual ring routing: network routing ispired by DHTs. In: Proceedings of ACM SIGCOMM'06 (2006)
17. Krioukov, D., Claffy, K.C., Fall, K., Brady, A.: On compact routing for the internet. In: Proceedings of ACM SIGCOMM'07 (2007)
18. Meyer, D., Zhang, L., Fall, K.: Report from the IAB workshop on routing and addressing. IETF RFC 4984
19. Jokela, P., Zahemszky, A., Esteve, C., Arianfar, S., Nikander, P.: LIPSIN: line speed publish/subscribe inter-networking. In: Proceedings of ACM SIGCOMM'09 (2009)
20. FIA 2010 conference report. http://www.future-internet.eu/fileadmin/documents/valencia_documents/FIA_Valencia_Report_v3_0__out_final_0306.pdf
21. Turner, J., Taylor, D.: Diversifying the internet. IEEE GLOBECOM (2005)
22. Openflow consortium. http://www.openflowswitch.org/
23. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.: The many faces of publish-subscribe. In: ACM Computing Surveys (CSUR) **35**(2) (2003)