

IFIP AICT 473



Sheikh Mahbub Habib
Julita Vassileva
Sjouke Mauw
Max Mühlhäuser
(Eds.)

Trust Management X

10th IFIP WG 11.11 International Conference, IFIPTM 2016
Darmstadt, Germany, July 18–22, 2016
Proceedings

 Springer

Editor-in-Chief

Kai Rannenber, Goethe University Frankfurt, Germany

Editorial Board

Foundation of Computer Science

Jacques Sakarovitch, Télécom ParisTech, France

Software: Theory and Practice

Michael Goedicke, University of Duisburg-Essen, Germany

Education

Arthur Tatnall, Victoria University, Melbourne, Australia

Information Technology Applications

Erich J. Neuhold, University of Vienna, Austria

Communication Systems

Aiko Pras, University of Twente, Enschede, The Netherlands

System Modeling and Optimization

Fredi Tröltzsch, TU Berlin, Germany

Information Systems

Jan Pries-Heje, Roskilde University, Denmark

ICT and Society

Diane Whitehouse, The Castlegate Consultancy, Malton, UK

Computer Systems Technology

Ricardo Reis, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

Security and Privacy Protection in Information Processing Systems

Yuko Murayama, Iwate Prefectural University, Japan

Artificial Intelligence

Ulrich Furbach, University of Koblenz-Landau, Germany

Human-Computer Interaction

Jan Gulliksen, KTH Royal Institute of Technology, Stockholm, Sweden

Entertainment Computing

Matthias Rauterberg, Eindhoven University of Technology, The Netherlands

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the first World Computer Congress held in Paris the previous year. A federation for societies working in information processing, IFIP's aim is two-fold: to support information processing in the countries of its members and to encourage technology transfer to developing nations. As its mission statement clearly states:

IFIP is the global non-profit federation of societies of ICT professionals that aims at achieving a worldwide professional and socially responsible development and application of information and communication technologies.

IFIP is a non-profit-making organization, run almost solely by 2500 volunteers. It operates through a number of technical committees and working groups, which organize events and publications. IFIP's events range from large international open conferences to working conferences and local seminars.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is generally smaller and occasionally by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is also rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

IFIP distinguishes three types of institutional membership: Country Representative Members, Members at Large, and Associate Members. The type of organization that can apply for membership is a wide variety and includes national or international societies of individual computer scientists/ICT professionals, associations or federations of such societies, government institutions/government related organizations, national or international research institutes or consortia, universities, academies of sciences, companies, national or international associations or federations of companies.

More information about this series at <http://www.springer.com/series/6102>

Sheikh Mahbub Habib · Julita Vassileva
Sjouke Mauw · Max Mühlhäuser (Eds.)

Trust Management X

10th IFIP WG 11.11 International Conference, IFIPTM 2016
Darmstadt, Germany, July 18–22, 2016
Proceedings

Editors

Sheikh Mahbub Habib
Technische Universität Darmstadt
Darmstadt
Germany

Julita Vassileva
University of Saskatchewan
Saskatoon, SK
Canada

Sjouke Mauw
University of Luxembourg
Luxembourg
Luxembourg

Max Mühlhäuser
Technische Universität Darmstadt
Darmstadt
Germany

ISSN 1868-4238 ISSN 1868-422X (electronic)
IFIP Advances in Information and Communication Technology
ISBN 978-3-319-41353-2 ISBN 978-3-319-41354-9 (eBook)
DOI 10.1007/978-3-319-41354-9

Library of Congress Control Number: 2016942509

© IFIP International Federation for Information Processing 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

This volume contains the proceedings of the 10th Annual IFIP Working Group 11.11 International Conference on Trust Management (IFIP TM). This is an annual research conference, organized by the International Federation for Information Processing Working Group WG 11.11, which started in 2007. The previous editions were held in New Brunswick (Canada, 2007), Trondheim (Norway, 2008), West Lafayette (USA, 2009), Marioka (Japan, 2010), Copenhagen (Denmark, 2011), Surat (India, 2012), Malaga (Spain, 2013), Singapore (2014), and Hamburg (Germany, 2015). This year, IFIP TM was part of the “Security&Privacy Week” (SPW) in Darmstadt, where more than a handful of security and privacy conferences and workshops took place. IFIP TM 2016 and the SPW were hosted by the Technische Universität Darmstadt, Germany, during July 18–22, 2016.

IFIP TM is a flagship conference of the IFIP Working Group 11.11. It focuses on novel research topics related to computational trust and trust-related issues of security and privacy. The IFIP TM 2016 conference invited contributions in several areas, including but not limited to trust architecture, trust modeling, trust metrics and computation, reputation and privacy, security and trust, socio-technical aspects of trust, and attacks on trust and reputation systems.

This year, we received 26 submissions from different parts of the world, including Australia, Belgium, Canada, China, Colombia, Egypt, Germany, Greece, Hong Kong, India, Indonesia, Israel, Japan, Malaysia, The Netherlands, Norway, Singapore, Spain, UK, and the USA. Every submission went through a peer-review process, with at least three reviewers. After carefully analyzing all the reviews, we accepted seven full papers (acceptance rate of 26.92 %) in addition to seven short papers.

Every year IFIP TM hosts the William Winsborough Commemorative Address in memoriam of our esteemed colleague Prof. William Winsborough. The award is given to an individual who has significantly contributed to the areas of computational trust and trust management. In 2016, the Working Group was pleased to host Prof. Simone Fischer-Hübner of Karlstad University, Sweden, to present a keynote speech on “Transparency, Privacy and Trust Technology for Tracking and Controlling my Data Disclosures: Does this Work?” An invited paper related to the keynote is also included in the proceedings.

In addition to papers and the William Winsborough keynote address, IFIP TM hosted Prof. Vijay Varadharajan of Macquarie University Sydney, Australia, to present a keynote speech on “Trust Enhanced Secure Role-based Access Control on Encrypted Data in Cloud.” An abstract of his speech is also included in these proceedings. Finally, the conference hosted a special panel session on “The Ideology of Social Science Meets The Digitisation of Trust, Security and Privacy,” organized and chaired by Dr. Natasha Dwyer of Victoria University Melbourne, Australia, and Sarah Talboom of Vrije Universiteit Brussel, Belgium. This session is exclusively organized for the speakers of the accepted papers in order to let them share the stories behind their papers.

In order to organize a successful conference, a team of dedicated people is a key. We would like to thank our honorable Program Committee members as well as additional reviewers for their timely, insightful, and thoughtful reviews. We are also fortunate to get a professional and friendly team of workshop and tutorial, panel and special session, graduate symposium, Web and Publicity chairs, and local organization chairs. Since IFIP TM 2016 is part of the ‘Security&Privacy Week’, thanks and appreciation go to local organization team members, especially Verena Giraud and Matthias Schulz. Finally, thanks to the Technische Universität Darmstadt and the funded projects and centers such as CROSSING, the Doctoral School “Privacy and Trust for Mobile Users,” and CYSEC at TU Darmstadt for providing the facilities and financial support.

Authors are essential for the success of conferences. Congratulations to all of those who got accepted and thanks to those who submitted to become a part of this research community. A number of conferences are out there that have trust among their topics of interest. IFIP TM distinguishes itself with its focus on the application of computational models of trust and trust management in different fields such as cybersecurity, privacy, human–computer interaction, social sciences, and risk quantification. We strive to build IFIP TM as a cross-disciplinary conference and without your support and feedback this would be impossible.

For more information on the working group, please visit <http://www.ifiptm.org/>

We hope that you enjoyed the conference and reading the proceedings.

May 2016

Sheikh M. Habib
Julita Vassileva

IFIP Trust Management X

10th IFIP W.G. 11.11 International Conference on Trust Management, 2016

**Darmstadt, Germany
July 18–22, 2016**

General Chairs

Sjouke Mauw University of Luxembourg, Luxembourg
Max Mühlhäuser Technische Universität Darmstadt, Germany

Program Chairs

Sheikh Mahbub Habib Technische Universität Darmstadt, Germany
Julita Vassileva University of Saskatchewan, Canada

Workshop and Tutorial Chairs

Masakatsu Nishigaki Shizuoka University, Japan
Jan-Phillip Steghöfer Göteborg University, Sweden

Panel and Special Session Chairs

Natasha Dwyer Victoria University, Australia
Sarah Talboom Vrije Universiteit Brussel, Belgium

Graduate Symposium Chairs

Christian Jensen Technical University of Denmark
Stephen Marsh University of Ontario Institute of Technology, Canada

Web and Publicity Chair

Anirban Basu KDDI R&D Laboratories, Japan

Local Organization Chair

Sascha Hauke Technische Universität Darmstadt, Germany

Program Committee

Stephen Marsh	UOIT, Canada
Anirban Basu	KDDI R&D Laboratories, Japan
Audun Jøsang	University of Oslo, Norway
Christian Damsgaard Jensen	Technical University of Denmark, Denmark
Yuko Murayama	Tsuda College, Japan
Natasha Dwyer	Victoria University, Australia
Pierangela Samarati	Università degli Studi di Milano, Italy
Peter Herrmann	Norwegian University of Science and Technology, Norway
Fabio Martinelli	IIT-CNR, Italy
Carmen Fernández-Gago	University of Malaga, Spain
Günther Pernul	Universität Regensburg, Germany
Jie Zhang	Nanyang Technological University, Singapore
Zeinab Noorian	Ryerson University, Canada
Ehud Gudes	Ben-Gurion University, Israel
David Chadwick	University of Kent, UK
Masakatsu Nishigaki	Shizuoka University, Japan
Tim Muller	Nanyang Technical University
Sara Foresti	Università degli Studi di Milano, Italy
Roslan Ismail	Tenaga National University, Malaysia
Rehab Alnemr	HP Labs, Bristol, UK
Nurit Gal-Oz	Sapir Academic College, Israel
Simone Fischer-Hübner	Karlstad University, Sweden
Claire Vishik	Intel Corporation, UK
Sascha Hauke	Technische Universität Darmstadt, Germany
Jesus Luna Garcia	Cloud Security Alliance and TU Darmstadt, Germany
Yuecel Karabulut	Oracle, USA
Tim Storer	University of Glasgow, UK
Hui Fang	Shanghai University of Finance and Economics, China
Shouhuai Xu	University of Texas at San Antonio, USA
Babak Esfandiari	Carleton University, Canada
Tanja Ažderska	Jožef Stefan Institute, Slovenia
Gabriele Lenzini	University of Luxembourg, Luxembourg
Weizhi Meng	Institute for Infocomm Research (I2R), Singapore
Piotr Cofta	British Telecom, UK
Jetzabel Serna-Olvera	Goethe Universität Frankfurt, Germany
Felix Gomez Marmol	NEC Labs Europe, Germany

Additional Reviewers

Colin Boyd	Norwegian University of Science and Technology, Norway
Jenni Ruben	Karlstad University, Sweden
Dai Nishioka	Iwate Prefectural University, Japan
Christian Richthammer	Universität Regensburg, Germany
Johannes Säger	Universität Regensburg, Germany

Trust Enhanced Secure Role-based Access Control on Encrypted Data in Cloud (Abstract of Keynote Talk)

Vijay Varadharajan

Department of Computing
Faculty of Science
Macquarie University NSW 2109, Australia
vijay.varadharajan@mq.edu.au

Abstract. In this talk I will begin with a brief look at current trends in the technology scenery and some of the key security challenges that are impacting on business and society. In particular, on the one hand there have been tremendous developments in cyber technologies such as cloud, Big Data and Internet of Technologies.

Then we will consider security and trust issues in cloud services and cloud data. In this talk, we will focus on policy based access to encrypted data in the cloud. We will present a new technique, Role based Encryption (RBE), which integrates cryptographic techniques with role based access control. The RBE scheme allows policies defined by data owners to be enforced on the encrypted data stored in public clouds. The cloud provider will not be able to see the data content if the provider is not given the appropriate role by the data owner. We will present a practical secure RBE based hybrid cloud storage architecture, which allows an organisation to store data securely in a public cloud, while maintaining the sensitive information related to the organisation's structure in a private cloud.

Then we will consider trust issues in RBE based secure cloud data systems. We will discuss two types of trust models that assist (i) the data owners/users to evaluate the trust on the roles/role managers in the system as well as (ii) the role managers to evaluate the trust on the data owners/users for when deciding on role memberships. These models will take into account the impact of role hierarchy and inheritance on the trustworthiness of the roles and users. We will also consider practical application of the trust models and illustrate how the trust evaluations can help to reduce the risks and enhance the quality of decision making by data owners and role managers of the cloud storage services.

Contents

Willam Winsborough Award Invited Paper

Transparency, Privacy and Trust – Technology for Tracking and Controlling My Data Disclosures: Does This Work?	3
<i>Simone Fischer-Hübner, Julio Angulo, Farzaneh Karegar, and Tobias Pulls</i>	

Full Papers

How to Use Information Theory to Mitigate Unfair Rating Attacks	17
<i>Tim Muller, Dongxia Wang, Yang Liu, and Jie Zhang</i>	
Enhancing Business Process Models with Trustworthiness Requirements	33
<i>Nazila Gol Mohammadi and Maritta Heisel</i>	
A Model for Personalised Perception of Policies	52
<i>Anirban Basu, Stephen Marsh, Mohammad Shahriar Rahman, and Shinsaku Kiyomoto</i>	
Evaluation of Privacy-ABC Technologies - a Study on the Computational Efficiency	63
<i>Fatbardh Veseli and Jetzabel Serna</i>	
A Trust-Based Framework for Information Sharing Between Mobile Health Care Applications	79
<i>Saghar Behrooz and Stephen Marsh</i>	
Supporting Coordinated Maintenance of System Trustworthiness and User Trust at Runtime	96
<i>Torsten Bandyszak, Micha Moffie, Abigail Goldstein, Panos Melas, Bassem I. Nasser, Costas Kalogiros, Gabriele Barni, Sandro Hartenstein, Giorgos Giotis, and Thorsten Weyer</i>	
Limitations on Robust Ratings and Predictions	113
<i>Tim Muller, Yang Liu, and Jie Zhang</i>	

Short Papers

I Don't Trust ICT: Research Challenges in Cyber Security	129
<i>Félix Gómez Mármol, Manuel Gil Pérez, and Gregorio Martínez Pérez</i>	

The Wisdom of Being Wise: A Brief Introduction to Computational
Wisdom. 137
Stephen Marsh, Mark Dibben, and Natasha Dwyer

Trust It or Not? An Empirical Study of Rating Mechanism and Its Impact
on Smartphone Malware Propagation. 146
Wenjuan Li, Lijun Jiang, Weizhi Meng, and Lam-For Kwok

Towards Behavioural Computer Science 154
Christian Johansen, Tore Pedersen, and Audun Jøsang

Improving Interpretations of Trust Claims 164
Marc Sel

Trust and Regulation Conceptualisation: The Foundation for User-Defined
Cloud Policies 174
Jörg Kebedies, Felix Kluge, Iris Braun, and Alexander Schill

A Calculus for Distrust and Mistrust 183
Giuseppe Primiero

Author Index 191

**Willam Winsborough Award Invited
Paper**

Transparency, Privacy and Trust – Technology for Tracking and Controlling My Data Disclosures: Does This Work?

Simone Fischer-Hübner^(✉), Julio Angulo, Farzaneh Karegar,
and Tobias Pulls

Department of Computer Science, Karlstad University, Karlstad, Sweden
{simone.fischer-huebner, julio.angulo,
farzaneh.karegar, tobias.pulls}@kau.se

Abstract. Transparency is a basic privacy principle and social trust factor. However, in the age of cloud computing and big data, providing transparency becomes increasingly a challenge.

This paper discusses privacy requirements of the General Data Protection Regulation (GDPR) for providing ex-post transparency and presents how the transparency-enhancing tool Data Track can help to technically enforce those principles. Open research challenges that remain from a Human Computer Interaction (HCI) perspective are discussed as well.

Keywords: Privacy · Transparency · Transparency-enhancing tools · Usability

1 Introduction

Transparency is an important factor for establishing user trust and confidence, as trust in an application can be enhanced if procedures are clear, transparent and reversible, so that users feel in control [1, 19]. However, especially in the context of cloud computing and big data, end users are often lacking transparency, as pointed out by the Art. 29 Data Protection Working Party [4, 5].

Big data analyses practices raise concerns in regard transparency, as individuals, unless they are provided with sufficient information, are often subject to decisions that they do not understand nor have control over.

Moreover, cloud users and data subjects lack transparency in regard to the involved supply chain with multiple processors & subcontractors, different geographic locations within the EEA (European Economic Area), transfers to third-party countries outside the EEA, and how a cloud service reacts to requests for access to personal data by law enforcement. In addition, there is a lack of intervenability for individuals, as there is a lack of tools for them for exercising their data subjects' rights.

Empirical research conducted in the EU project A4Cloud¹ for eliciting cloud customer requirements revealed that cloud customers will increase their trust that their data is secure in the cloud, if there is transparency about what is possible to do with the data, possible exit procedures (“way out”) and the ownership of the data [16].

¹ EU FP7 project A4Cloud (Accountability for the Cloud), <http://www.a4cloud.eu/>.

Transparency of personal data processing is also an important principle for the individual's privacy as well as for a democratic society. As the German constitutional court declared in its Census Decision², a society, in which citizens could not know any longer who does when, and in which situations know what about them, would be contradictory to the right of informational self-determination. Consequently, the European Legal Data Protection Framework is granting data subjects information, access and control rights enforcing transparency and intervenability. Transparency-Enhancing Tools (TETs) can help to enable the individual's right for transparency also by technological means.

In this article, we discuss the data subject rights in regard to transparency and intervenability by the EU General Data Protection Regulation (GDPR [10]) (Sect. 2), and how they can be technically enforced by TETs and particularly by different versions and functions of the Data Track tool that has been developed at Karlstad University within the scope of the PRIME³, PrimeLife⁴ and A4Cloud EU projects (Sect. 3). We discuss HCI and trust challenges in regard to the Data Track (Sect. 4), related work (Sect. 5) and conclude with follow-up research questions (Sect. 6).

2 Transparency

The concept of transparency comprises both 'ex ante transparency', which enables the anticipation of consequences before data are actually disclosed (e.g., with the help of privacy policy statements), as well as 'ex post transparency', which informs about consequences if data already have been revealed (e.g., what data are processed by whom and whether the data processing is in conformance with negotiated or stated policies) [15].

The EU General Data Protection Regulation, which is likely to be enacted in the first half of 2016, comprises different data subject rights for providing both ex ante and ex post transparency as well as means for intervenability and control, which are extending the fundamental rights of data subjects that were provided by the EU Data Protection Directive 95/46/EC [9].

Ex ante Transparency. Ex ante transparency is a condition for data subjects of being in control and for rendering a consent⁵, which has to be informed, valid.

Pursuant to Art 14 GDPR, the data controller must ensure that the data subject is provided with required privacy policy information at the time when the data is collected from the data subject, including information about the identity of the data controller and

² German Constitutional Court, Census decision ("Volkszählungsurteil"), 1983 (BVerfGE 65,1).

³ EU FP6 project PRIME (Privacy and Identity Management for Europe), <https://www.prime-project.eu/>.

⁴ EU FP7 project PrimeLife (Privacy and Identity Management for Europe for Life), <http://primelife.ercim.eu/>.

⁵ 'The data subject's consent' is defined by the Data Protection Directive as "any freely given specific and informed indication of his wishes by which the data subject signifies his agreement to personal data relating to him being processed".

the data processing purposes, and for ensuring fair and transparent processing also information about recipients/categories of recipients, intention to transfer data to a recipient in a third country or international organization, data subject rights incl. the right to withdraw consent at any time and the right to lodge complaint with supervisory authority, the legal basis and whether the data subject is obliged to provide the data and consequences of not providing the data, as well as the existence of automated decision making including profiling, the logic involved, significance and envisaged consequences.

Ex ante TETs include policy tools and languages, such as P3P [28] the PrimeLife Policy Language PPL [25] or A-PPL [6], which can help to make the core information of privacy policies and information on how far a services side's policy complies with a user's privacy preferences more transparent to an end user at the time when he is requested to consent to data disclosure.

Ex post Transparency and Intervenability. The GDPR provides data subjects with the right of access to their data pursuant to Art 15, which comprises the right to information about the data being processed, data processing purposes, data recipients or categories of recipients, as well as information about the logic involved on any automatic processing including profiling. In extension to the EU Data Protection Directive, data subjects should also be informed about the significance and envisaged consequences of such processing, as well as about safeguards taken in case of transfer to a third country. Another new provision of the GDPR for increasing transparency demands that the controller shall provide a copy of his/her personal data undergoing processing to the data subject, and if the data subject makes the request in electronic form, the information should be provided in “*an electronic form, which is commonly used*”.

Furthermore, the newly introduced right to Data Portability (Art.18), which is the right to receive data in a structured and commonly used and machine-readable format and the right to transmit it to another controller (or to have it transmitted directly from controller to controller). It can thus also be used as a means for enhancing transparency, even though its objective is to prevent that data subjects are “locked” into privacy-unfriendly services by allowing them easily to change providers along with their data. However, in contrast to the electronic copy of the data under processing that the data subject has the right to receive pursuant to Art. 15, exported data may only contain the data that the data subject explicitly or implicitly disclose, but not data that the service provider derived from that data, as such derived data (e.g., in the form of user profiles) may comprise business value for a company and a transfer to a competing service provider would thus have a strong impact on that company.

This data subject right that is providing ex post transparency is also a prerequisite for exercising the data subject rights to withdraw consent at any time, which should be made as easy as to give it (Art. 5), to obtain the correction or deletion, the right to restrict the processing as well as the newly introduced right to be forgotten in a timely manner (Art. 16, 17, 17a).

In addition to the transparency rights in the GDPR, specific ex post transparency rights are, for instance, provided by the Swedish Data Patient Act [27] to data subjects by requiring that health care providers have to inform patients upon request about who has accessed their medical information.

In the next section, we will discuss how the subsequent version of the Data Track can empower users to exercise these ex post transparency rights.

3 The Data Track

The Data Track is a user side ex post transparency tool, for which different versions with subsequent enhancements have been developed within the EU research projects PRIME (FP6), PrimeLife (FP7), and A4Cloud (FP7).

PRIME and PrimeLife Data Track. The first version developed within the PRIME project includes a history function (see [23]), which was later complemented in the PrimeLife project with online access functions. The history function stores in a secure manner for each transaction, in which a user discloses personal data to a service, a record for the user on which personal data were disclosed to whom (i.e. the identity of the controller), for which purposes and, more precisely, under which agreed-upon privacy policy the user has given his/her consent, as well as a unique transaction ID. These records of consents can serve users as a reference for exercising his or her right to easily revoke consent at any time. The data disclosures are tracked by a middleware called the PRIME Core, running both on the user's side and at the remote service.

For exercising his or her rights to access, correct, delete or block data, the user needs to prove that he or she is the respective data subject. This can be done by proving knowledge of a unique transaction ID, which is stored in his/her Data Track and at the services' side for each transaction of personal data disclosure. Notably, for authentication, the data subject does not have to disclose any more personal data than what the service already knows. This allows in principle also anonymous or pseudonymous users to access their data at the services' side.

These records of provided consent stored in the user's Data Track can serve users as a reference for exercising his or her right to easily revoke consent at any time.

The Data Track's user interface version developed under the PrimeLife EU FP7 project provided search functions for the locally stored Data Track records as well as online access functions, which allowed users to easily get a tabular overview about what data they have disclosed to a services side and what data are still stored by the services' side, or what data have been inferred and added. This should allow users to check whether data have been changed, processed, added or deleted (and whether this was in accordance with the agreed-upon privacy policy).

Complete descriptions of the Data Track proof-of-concept and user interfaces developed under the PrimeLife project can be found in [29]. Usability tests of early design iterations of the PrimeLife's Data Track revealed however that many test users had problems to understand whether data records were stored in the Data Track client on the users' side (under the users' control) or on the remote service provider's side [11].

A4Cloud Data Track. Within the scope of the A4Cloud project, we have for developed and tested in several iterations alternative user interfaces (UIs) and HCI concepts consisting of graphical UI illustrations of where data are stored and to which entities data have been distributed. Graphical illustrations of data storage and data flows have a potential to display data traces more naturally as in real world networks.

Moreover, previous research studies suggest that network-like visualizations provide a simple way to understand the meaning behind some types of data [7, 13] and other recent studies claim that users appreciate graphical representations of their personal data flows in forms of links and nodes [17, 18].

Therefore, for the A4Cloud Data Track (also called “*GenomSynlig*”), we developed the so-called “*trace view*” (see Fig. 1), presenting an overview of which data items have been sent to service providers, as well as which service providers have received what data items about the user.

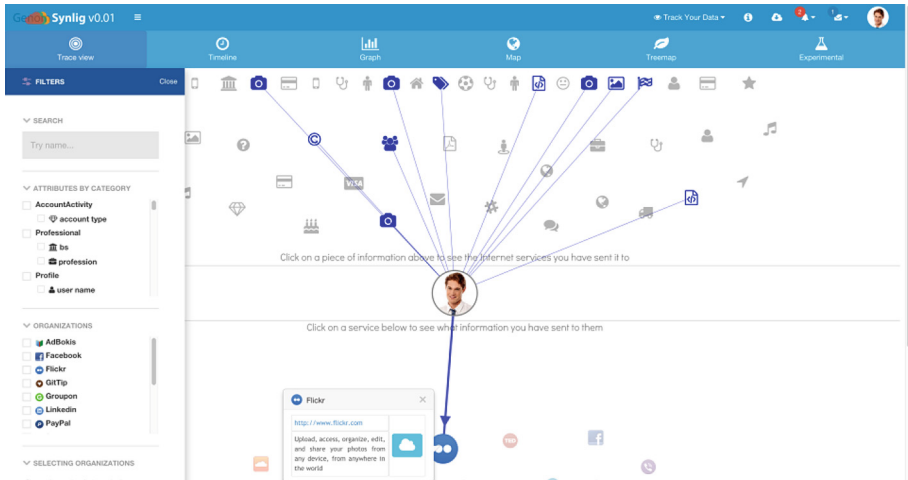


Fig. 1. The trace view user interface of the data track

The idea is that users should be able to view what selected personal data items stored in the Data Track (displayed by icons in the top panel of the UI) that they have submitted to services on the Internet (that are shown in the bottom panel of the interface). The user is represented by the panel in the middle by giving him or her the feeling that the Data Track is a user-centric tool.

If users click on one or many Internet service icons, they will be shown arrows pointing to the icons symbolising data items that those services have about them; in other words they can see a *trace* of the data that services have about them. Similarly, if they select icons of one or many data items (on the top), they will be shown arrows pointing to the Internet services that have received those data items.

In addition to this “*local view*” of the trace view, which is graphically displaying the information that is stored locally in the Data Track about what data has been disclosed to whom, a user can also exercise online access functions by clicking on the cloud icon next to the service provider’s logo, and see “*remote views*” in a pop-up window (see Fig. 2) what data the service provider has actually stored about him, which it either received explicitly or implicitly from the user or derived about him.

Clicking on the pencil or trash bin icons located right to the data items will activate functions for requesting correction or deletion of data at the services side.

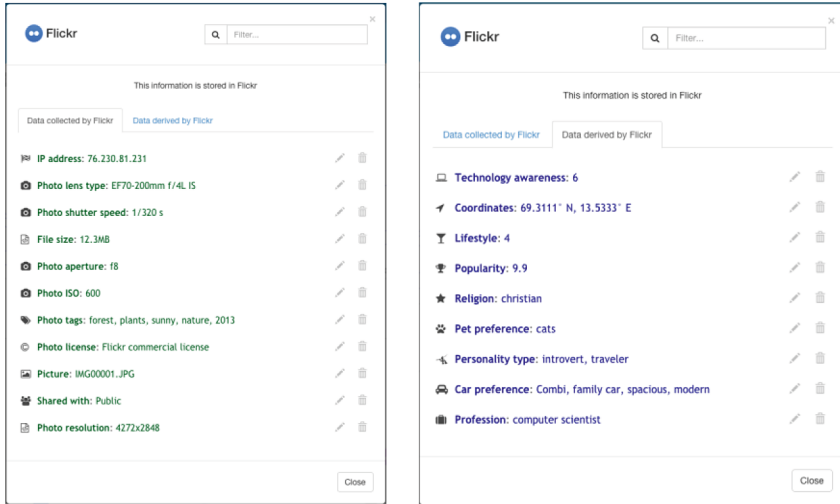


Fig. 2. Remote views of data stored at the services side that was either collected from the user (shown in the left side pop-up window) or derived about him (shown in the right side pop-up window).

An alternative *timeline view* has been developed as well for the Data Track, which lists the information about data disclosures in the Data Track records in chronological order for selected time intervals (see Fig. 3).

Within the scope of A4Cloud, a cryptographic system for performing privacy-preserving transparency logging for distributed systems (e.g., cloud-based systems) has been developed [26]. In combination with the transparency logging, the Data Track could also visualise personal data flows along a cloud chain.

At the end of the A4Cloud project, we developed an open source and standalone version of the Data Track that allows the visualisation of data exported from the Google Takeout service. We focused on the Google location history, as part of the Takeout data, and developed an additional a graphical map view to complement the trace view and timeline view. As depicted in Fig. 4, the map view allows to visualize location, activity and movement patterns as described in the location history provided by Google. Notably, activities are data derived by Google based on primarily the location reported by Android devices.

Table 1 provides an overview of the functions of the different Data Track versions and functions and the legal privacy principles that they address pursuant to the GDPR. It shows that the different functions of the Data Track that we have developed in the subsequent versions are complementing each other, as they address different legal privacy requirements for enabling transparency and intervenability.

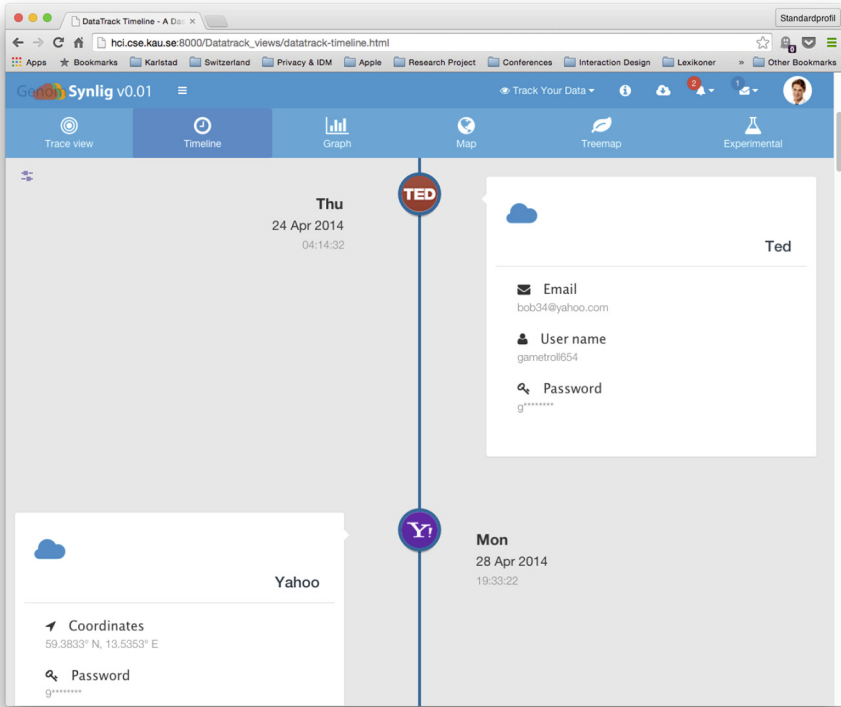


Fig. 3. The timeline view showing data disclosures in chronological order.

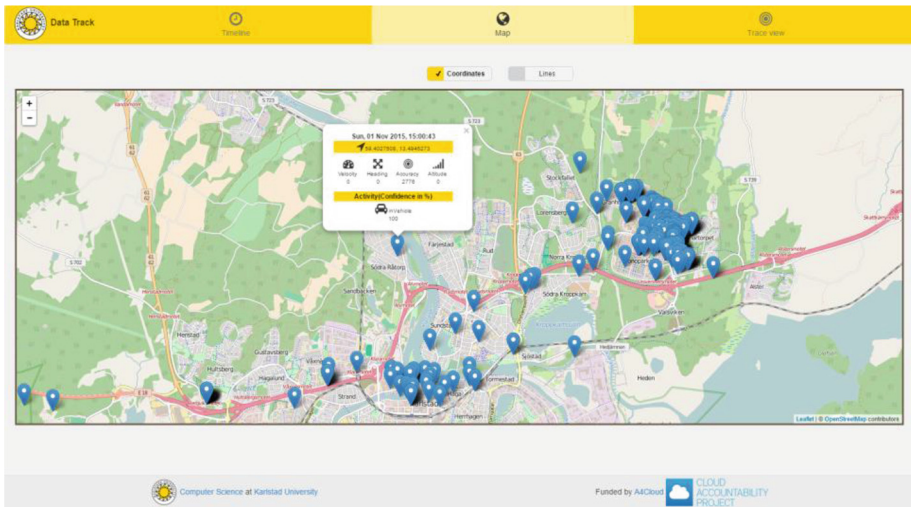


Fig. 4. The map view showing data locations, activities and movement patterns.

Table 1. Data track versions, functions and GDPR legal privacy principles addressed for achieving transparency and intervenability.

Version	Functions	GDPR Legal Principles addressed
PrimeLife Data Track [11, 29]	Local database of data disclosed, transaction pseudonyms, consent given Online access functions UI: Tabular Form	Consent Management – helps to enforce the right to object/revoke consent, pursuant to Art. 5 (Electronically provided) Data subject access functions (Art. 16, 17, 17a)
A4Cloud Data Track (“GenomSynlig”) [3, 8]	Local database of data disclosed, Consent given Online access functions Graphical UI: Trace View & Time line, search functions and tutorials	(as PrimeLife Data Track)
A4Cloud – Export Data Track Stand-alone Version ^a	Local visualisation of data exports UI: Additional graphical map view	Transparency of personal “big data” retrieved via data portability (Art. 18) or electronic copy of data (Art. 15) functions

^a<https://github.com/pylls/datatrack>

4 HCI Evaluation and Challenges

As pointed out in [22], the legal privacy principles, such as transparency principle, have HCI (Human Computer Interaction) implications as “they describe mental processes and behaviour that the data subjects must experience in order for a service to adhere to these principles”. In particular, the transparency principles requires that data subjects comprehend the transparency and control options, are aware of when they can be used, and are able to use them. Therefore, another important design criterion for TETs is usability.

Throughout the A4Cloud project, the user interface of the Data Track has gone through three iterations of design and user evaluations with 13-16 test participants in each iteration. The evaluations were performed at Karlstad University’s Ozlab for an e-Shopping scenario and consisted of a mixture of a user-based cognitive walk and a talk-aloud protocol, with which participants were encouraged to express their opinions and understanding aloud, followed by a post-test questionnaire. The evaluations had not only the objective of testing the level of comprehension of the interface, but was also a method for gathering end-user requirements on the needs and expectations that such a tool should provide to its users. Details about the results of the test iterations are reported in [2, 3, 8, 12].

In general, evaluations have also shown that participants understand the purpose of the tool and ways to interact with it, identifying correctly the data that has been sent to particular service providers, and using the filtering functions to answer questions about their disclosed personal data. The set of search functions provided for the last Data Track iteration led generally to better tracking results. Throughout the test iterations, the majority of test users also saw the Data Track as a potentially useful tool and

appreciated its transparency options and would use it on a regular basis. Most test users of the last test iteration preferred the trace view over the timeline view.

Also at an evaluation workshop organised by A4Cloud partner SINTEF, the advantages and possible risks of using a tool such as the Data Track were discussed, as well as the requirements to make such a tool not only usable but also adopted in their daily Internet activities. It was for instance commented by one participant that transparency provided by the Data Track, would encourage service providers to comply with their policies and be responsible stewards of their customers data, “*it would keep me informed and hold big companies in line*”. Another participant mentioned as a benefit the increased awareness of disclosures made to service providers, “*makes you aware of what information you put on the Internet, you probably would be more careful*” (see [16]).

Usability tests of earlier designs of the Data Track already revealed that users expressed feelings of surprise and discomfort with the knowledge that service providers analyse their disclosed data in order to infer additional insights about them, like for instance their music preferences or shopping behavior. Hence, making data processing practices for user profile should be an important functionality of an ex post TET.

The tests also revealed that there remain still difficulties for a larger portion of users to differentiate the local from the remote view, i.e. to differentiate between what data is locally stored under their control on their computers (shown by the trace or timeline view) and what data is stored on the services’ side but accessible via the online access functions shown through the pop-up dialog).

Some test users as also voiced scepticism of the level of security of their data. The Data Track storing big personal data becoming a single point of failure was also mentioned as a potential risk by participants of the SINTEF workshop [16].

Understanding that the data stored in the Data Track are under the user’s control, is however an important prerequisite for end user trust and adoption, along with effective means of security that are communicated to the users.

Security for the A4Cloud Data Track mainly relies on encryption (data is encrypted at rest). To avoid risks associated with long-term collection and central storage of personal data in the Data Track, the latest standalone version of the Data Track takes a different approach. Since the primary purpose of the standalone Data Track is to visualise data exported from online services, there is no need for long-term local storage. Once the Data Track is closed, all data collected locally is deleted together with the ephemeral encryption key that was used to temporarily store data while the Data Track was running.

However, results from first usability tests conducted on latest stand-alone Data Track version with 16 test users revealed once more the problem that test participants had problems to differentiate between what data was under their control (after they exported the data to their computers) and to what data the controller (in this case Google) still had access. Several of the test users did not understand that their exported location data that was visualised with the Data Track was a local copy stored on the user’s machine, but rather got the impression that the exported data was synchronized with Google’s remote data storage. Consequently, the idea behind deleting all exported data after closing the Data Track was not well understood by them.

5 Related Work

Related data tracking and control tools for end users are in contrast to the Data Track usually restricted to specific applications, cannot be used directly to track data along cloud chains or are not under complete control of the users. Examples are Mozilla's Lightbeam [21] that uses interactive visualizations to show the first and third party sites that a user is interacting with on the Web, and Google Dashboard [14], which grants its users access to a summary of the data stored with a Google account including account data and the users' search query history. In contrast to the Data Track, the Dashboard provides access only to authenticated (non-anonymous) users.

Related to the Data Track are services that are targeting at giving users back control of their own data, such as datacoup.com, as well as personal data vaults, such as [20] developed for participatory sensing applications, which includes a logging functionality that allows displaying transactions and transformations of users' data and enables users to track who has accessed their data.

The DataBait tool [24], developed within the EU FP7 research project USEMP⁶, allows users of online social networks to share their data with a secured trusted research platform, which uses Machine Learning algorithms to provide profile transparency by explaining how users may be targeted on the basis of their postings and behavioural data. In contrast to the Data Track, its emphasis has not been put on graphical visualisation of data traces. Besides, requires end users to entrust all their data to a transparency service operated by a third party, while the Data Track is a user-side tool that allows the user to keep complete control over the Data Track data. While user control is advantageous from a privacy perspective, it does however also put higher demands on the end users for setting up and running the Data Track in a safe system environment.

6 Conclusions and Outlook

Transparency is a basic privacy principle and social trust factor. In this paper, we discussed legal principles pursuant to the GDPR for providing transparency and intervenability for users and discussed how these principles can be enforced by TETs, and particularly by the Data Track as an example of an ex-post TET that can operate under complete user control. We show that the different Data Track functions that we have developed for the different Data Track versions are complementing each other, because they are addressing different legal privacy requirements of the GDPR for enabling ex-post transparency and intervenability.

While several iterations of usability tests have shown that end users appreciate the transparency functionality of the Data Track, the user's perception of control and security in regard to the Data Track remain challenges to be tackled for also promoting end user trust in the Data Track.

⁶ EU FP7 project USEMP (User Empowerment for enhanced Online Management), <http://www.usemp-project.eu>.

Further research challenges that we would like to tackle in our future research relate to transparency about the consequences of potential big data profiling by both service providers and other government agencies, such as for instance tax authorities conducting social network analyses for detecting tax fraud. In particular, we are interested to analyse how the right to data portability and/or the right to receive an electronic copy of one's data together with the right to information about the logic involved in profiling, can enable citizens to aggregate their data and to infer and understand what government applications might deduce from them via profiling and what the possible consequences can be.

References

1. Andersson, C., Camenisch, J., Crane, S., Fischer-Hübner, S., Leenes, R., Pearson, S., Pettersson, J.S., Sommer, D.: Trust in PRIME. In: Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology. IEEE Xplore (2005)
2. Angulo, J., Fischer-Hübner, S., Pettersson, J.S.: General HCI principles and guidelines for accountability and transparency in the cloud. A4Cloud deliverable D:C-7.1, A4Cloud Project, September 2013
3. Angulo, J., Fischer-Hübner, S., Pulls, T., Wästlund, E.: Usable transparency with the data track: a tool for visualizing data disclosures. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems – CHI 2015, pp. 1803–1808. ACM (2015)
4. Art. 29 Data Protection Working Party. Opinion 5/2012 on Cloud Computing. European Commission, 1 July 2012
5. Art. 29 Data Protection Working Party, Opinion 03/2013 on Purpose Limitation. European Commission, 2 Apr 2013
6. Azraoui, M., Elkhiyaoui, K., Önen, M., Bernsmed, K., De Oliveira, A.S., Sendor, J.: A-PPL: an accountability policy language. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/SETOP/QASA 2014. LNCS, vol. 8872, pp. 319–326. Springer, Heidelberg (2015)
7. Becker, R.A., Eick, S.G., Wilks, A.R.: Visualizing network data. *IEEE Trans. Vis. Comput. Graph.* **1**(1), 16–28 (1995)
8. Bernsmed, K., Fischer-Hübner, S., et al.: A4Cloud Deliverable D.D-5.4 User Interface Prototypes, 31 Sept 2015
9. European Commission. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Off. J. L.* **281**, 0031–0050, 23 Nov 1995
10. European Commission. Proposal for a Regulation of the European Parliament and of the Council on the protection of individual with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Brussels, 15 December 2015
11. Fischer-Hübner, S., Hedbom, H., Wästlund, E.: Trust and assurance HCI. In: Camenisch, J., Fischer-Hübner, S., Rannenberg, K. (eds.) *Privacy and Identity Management for Life*, pp. 245–260. Springer, Heidelberg (2011)

12. Fischer-Hübner, S., Angulo, J., Pulls, T.: How can cloud users be supported in deciding on, tracking and controlling how their data are used? In: Hansen, M., Hoepman, J.-H., Leenes, R., Whitehouse, D. (eds.) *Privacy and Identity 2013*. IFIP AICT, vol. 421, pp. 77–92. Springer, Heidelberg (2014)
13. Freeman, L.C.: Visualizing social networks. *J. Soc. Struct.* **1**(1), 4 (2000)
14. Google. Google dashboard. <https://www.google.com/settings/dashboard>
15. Hildebrandt, M.: Behavioural biometric profiling and transparency enhancing tools. FIDIS Deliverable, D7.12. FIDIS EU project (2009)
16. Jaatun, M.G., Cruzes, D.S., Angulo, J., Fischer-Hübner, S.: Accountability through transparency for cloud customers. In: Helfert, M., Muñoz, V.M., Ferguson, D. (eds.) *Cloud Computing and Services Science*, pp. 38–57. Springer International Publishing, Switzerland (2015)
17. Kani-Zabihi, E., Helmhout, M., Coles-Kemp, L.: Increasing service users’ privacy awareness by introducing on-line interactive privacy features. In: *IAAC Symposium 2011* (2012)
18. Kolter, J., Netter, M., Pernul, G.: Visualizing past personal data disclosures. In: *International Conference on Availability, Reliability, and Security, ARES 2010*. IEEE (2010)
19. Lacochee, H., Crane, S., Phippen, A.: *Trustguide: Final Report* (2006)
20. Maguire, M., Bevan, N.: User requirements analysis. In: Mun, M. Hao, S., Mishra, N., Shilton, K., Burke, J., Estrin, D., Hansen, M., Govindan, R. (eds.) *Proceedings of IFIP 17th World Computer Congress. Personal Data Vaults: a Locus of Control for Personal Data Streams, CoNEXT 2010*: 17. ACM Digital Library (2002)
21. Mozilla. Lightbeam add-on for Firefox. <https://www.mozilla.org/en-US/lightbeam/>
22. Patrick, A.S., Kenny, S.: From privacy legislation to interface design: implementing information privacy in human-computer interactions. In: Dingledine, R. (ed.) *PET 2003*. LNCS, vol. 2760, pp. 107–124. Springer, Heidelberg (2003)
23. Pettersson, J.S., Fischer-Hübner, S., Bergmann, M.: Outlining “data track”: privacy-friendly data maintenance for end-users. In: Wojtkowski, W., Wojtkowski, W.G., Zupancic, J., Magyar, G., Knapp, G. (eds.) *Advances in Information Systems Development*, pp. 215–226. Springer US, Heidelberg (2007)
24. Popescu, A., et al.: User empowerment for enhanced online presence management – use cases and tools. In: *Amsterdam Privacy Conference 2015*, pp. 23–26, Amsterdam, 8 October 2015
25. PrimeLife, Privacy and Identity Management in Europe for Life - Policy Languages. <http://primelife.ercim.eu/results/primer/133-policy-languages>
26. Pulls, T., Peeters, R., Wouters, K.: Distributed privacy-preserving transparency logging. In: *Workshop on Privacy in the Electronic Society*. ACM (2013)
27. Svensk Författningssamling Riksdagen. Patientdatalag (2008: 355)
28. W3C, “P3P – The Platform for Privacy Preferences 1.1 (P3P1.1) Specification”, W3C Working Group Note, 13 November 2006. <http://www.w3.org/P3P/>
29. Wästlund, E., Fischer-Hübner, S.: End user transparency tools: UI prototypes. PrimeLife Deliverable D.4.2.2. PrimeLife project (2010)

Full Papers

How to Use Information Theory to Mitigate Unfair Rating Attacks

Tim Muller^(✉), Dongxia Wang, Yang Liu, and Jie Zhang

Nanyang Technological University, Singapore, Singapore
t.j.c.muller@gmail.com

Abstract. In rating systems, users want to construct accurate opinions based on ratings. However, the accuracy is bounded by the amount of information transmitted (leaked) by ratings. Rating systems are susceptible to unfair rating attacks. These attacks may decrease the amount of leaked information, by introducing noise. A robust trust system attempts to mitigate the effects of these attacks on the information leakage. Defenders cannot influence the actual ratings: being honest or from attackers. There are other ways for the defenders to keep the information leakage high: blocking/selecting the right advisors, observing transactions and offering more choices. Blocking suspicious advisors can only decrease robustness. If only a limited number of ratings can be used, however, then less suspicious advisors are better, and in case of a tie, newer advisors are better. Observing transactions increases robustness. Offering more choices may increase robustness.

1 Introduction

Online systems nowadays are typically too large for a single user to oversee. A user must rely on recommendations, reviews, feedback or *ratings* from other users (i.e., advisors), to be able to use a system to its fullest extent. In practice, we see that ratings are ubiquitous in large online systems. The exact design challenges introduced by supporting ratings depend on context (e.g. rating format, distributing ratings, subjectivity). One major challenge for all systems is how to deal with unfair ratings.

Typical approaches perform some or all of the following: incentivise honest ratings, detect and filter unfair ratings, update advisors' trustworthiness. More involved approaches may attempt to use possibly unfair ratings, and correct for the possible error, e.g. using machine learning or statistical methods. We call such methods aggregation mechanisms. The power of aggregation mechanisms is limited. Specifically, given a set of ratings, the amount of information that can be extracted is bounded upwards by a certain quantity. We call this quantity the *information leakage* of the ratings. No aggregation mechanism can be expected to do better than that.

Fortunately, the set of ratings that an aggregation mechanism operates on is not a universal given. One may control factors such as the number of advisors, which advisors to ask and the rating format. Changing these factors will

© IFIP International Federation for Information Processing 2016

Published by Springer International Publishing Switzerland 2016. All Rights Reserved
S.M. Habib et al. (Eds.): IFIPTM 2016, IFIP AICT 473, pp. 17–32, 2016.

DOI: 10.1007/978-3-319-41354-9_2

change the information leakage of the ratings, and thus the limits of the power of aggregation mechanisms. Ideally, we want to increase the limit.

We formalise an abstract model of rating systems that assumes the bare minimum. Its only assumption is that there exist honest advisors, and their ratings correlate somehow with the decision that a user should make. Furthermore, we show that information theory offers effective tools to measure the quality of ratings. Specifically, we prove that information leakage of a rating about a decision puts a hard bound on the accuracy of a decision. The remaining results are a set of design guidelines. These guidelines are valid for any rating system that has ratings that somehow correlate with good decisions. Specifically, (1) blocking suspicious advisors is not helpful and can decrease robustness, (2) when receiving ratings is costly, less suspicious advisors should be preferred, (3) and if advisors are equally suspicious, newer ones are preferable, (4) if possible, keep track of who has direct experience, and (5) changing the rating format and the options in a decision may increase robustness.

The paper is organised as follows: we discuss related idea and approaches in Sect. 2. We discuss the problem at hand – unfair rating attacks, in Sect. 3. Then we introduce the abstract notion of trust systems, in Sect. 4, formalise them (Sect. 4.1) and discuss what defenders can alter (Sect. 4.2). Then we show that limiting the information leakage is limiting the accuracy, in Sect. 5. In Sect. 6, we prove the five aforementioned guidelines.

2 Related Work

Multiple types of approaches exist to deal with unfair rating attacks. Some approaches provide incentives to promote honest rating behaviour [5, 6, 18]. Jurca and Faltings design a payment-based incentive scheme, which explicitly rewards honest feedback by an amount that offsets both the cost and the benefit of lying [5]. The payment schemes can be based on proper scoring rules, or correlation between the ratings of different advisors. In [6], they study how to resist against collusive advisors: colluders that share a lying strategy have to suffer monetary losses. Some other approaches aim to detect and filter out unfair ratings [15, 16]. For product-rating based online rating systems, Yafei et al. propose to detect collaborative biased ratings by observing time intervals where they are highly likely [16]. Reporting ratings is treated as a random process, and signal-processing techniques are applied to detect changes in rating values (e.g., detecting mean change). Most approaches evaluate advisors’ trustworthiness, based on which reliable advisors are selected or ratings get discounted [14, 17]. Yu et al., propose a reinforcement learning based framework to filter unfair ratings and make more accurate decisions in selecting trustees [17]. Both direct experiences and indirect evidences from advisors are aggregated to select highly reputable trustees. The reward derived from the interaction with a trustee is used to update advisors’ credibility, and ratings from less credible advisors are discarded. Meanwhile, weights assigned to direct and indirect trust evidences are also updated in trust evidence aggregation. We call these defense approaches as aggregation mechanisms.

The classification for different aggregation mechanisms is not absolute. Different types of approaches may be aggregated. For example, in [16], statistical methods are used to detect unfair ratings, of which the results are used to evaluate trustworthiness of advisors. The trustworthiness of advisors is then used to aggregate ratings, and also detect future suspicious ratings.

Despite of deviating from the truth, unfair ratings may still contain useful information (e.g., if they are correlated with a user’s direct experiences). There are approaches which exploit such correlation to make use of unfair ratings [9–11]. BLADE [9] and HABIT [10] learn from statistical correlations between a user’s direct experiences and an advisor’s ratings to adjust his ratings. For example, if an advisor always report bad ratings about a trustee, of which the user has good trust opinion, then his ratings get reversed. In this paper, we proved that suspicious advisors may still provide useful ratings (Proposition 4). Ratings from honest advisors may be subjectively different from a user’s direct experiences, but they differentiate from unfair ratings from attackers. Subjective ratings may provide useful information as they are relevant for a user. By directly discarding or filtering ratings that deviate from direct experiences, subjective ratings from honest advisors may also get excluded.

The quantification of the amount of information in ratings (i.e., *information leakage*) is already well studied in [12]. The defense approaches above cannot change the information leakage of ratings in a system, and they only differ in the way of exploiting it. Hence, their effectiveness is limited. From [12], we know that different attacks make information leakage in a system different. In the worst-case attacks where there is little information leakage, these approaches may not help at all. A robust rating system should not let its limitation to be controlled by attacks. Hence, it is vital to increase the limit of information leakage under attacks.

We found that some properties of a system, like the format of ratings, can affect the information leakage. Also, the conditions to achieve the minimal information leakage may also change based on these properties. By proper design, the power of defense approaches can be limited less, and the power of attacks can be decreased.

3 Unfair Rating Attacks

Unfair rating attacks are known to exist. They have been detected on existing trust systems [3,7], and they are well-studied in the literature [4]. It seems straightforward what an unfair rating attack is (unfair ratings are provided to mislead a user). But in reality, only ‘rating’ is unambiguous. For example, subjectivity may blur the truth and lies, meaning ratings deviating from the truth may not be from attackers, but subjective honest advisors. Moreover, with some probability, an honest user may perform the same sequence of actions (trace) as a user that intends to attack the system [2]; is that trace an attack? The issues lies in considering only the actual ratings.

Ratings cannot be fair or unfair by themselves. Not only may subjectivity lead to false ratings that are not unfair, but unfair ratings can be (objectively or

subjectively) true. Advisors may tell the truth to mislead users that believe the advisor is more likely to lie [11]. Advisors may tell the truth because they are colluding, but want to remain undetected [12]. Or advisors may tell the truth because they do not want to lose the user’s trust [13]. In each case, the unfairness lays in the fact that the advisor merely acts out some malicious strategy, rather than respecting the truth.

We want to have a pragmatic definition of unfair rating attacks. Our goal is to make rating systems robust against unfair rating attacks. In other words, ratings must be useful, even if some sources are malicious. However, how useful ratings are to a user, depends on what the user chooses to do with these ratings. The aim of this paper is not to provide the right aggregation mechanism or dictate user’s decisions, so – pragmatically – we take a measure of how much a user can do with the ratings: information leakage. We prove, in Theorem 1, that the information leakage measures the potential usefulness of ratings.

Attackers have an underlying strategy. Attacks are considered successful, when they achieve some goal. The goal is not known in advance. Since we are considering robustness, we primarily care about the worst-case for a user – the information leakage is minimal. Hence, we pragmatically assert that the goal of an attack is to minimise information leakage. We assume attackers select the strategy that minimises information leakage. If we are wrong, then the information leakage increases by definition. Section 5.1 provides detailed formal analysis.

4 Rating System

In this paper, we are not necessarily interested in rating systems themselves, but rather in which aspects we can control in our advantage. Particularly, we study how to set the parameters that we control to harden a rating system – maximising the minimal information leakage (Sect. 3). In this section, we present an abstract representation of a rating system, that allows us to analyse the relevant aspects without dissolving in details.

Users make decisions based on ratings. Some decisions are better than others. We take a simple notion of correctness of decisions. Users are given n choices to make a decision, of which 1 choice is the best option. The relevant part is that we have a ground truth, that the user wants to deduce.

We exemplify the theory with a simple example. The results that we present in this paper are general results for all rating systems that follow the notions from this section. However, the results are easier to interpret on a simple example (e-commerce) system:

Example 1. On an e-commerce system, two similar products are offered for sale. The user has three options: buy product x , buy product y , or buy neither. In the e-commerce system, another buyer can fulfill the role of advisor, and assign scores of 1–5 stars to x , y , neither or both. Each of these (combinations of) scores may imply something about the user’s decision, hence the abstract rating must contain each. Thus, there are 1 (neither) plus 5 (just x) plus 5 (just y) plus $5 \cdot 5$ (both), which is 36, possible ratings. An honest advisor provides a rating that –

at the very least – correlates with the correct decision. Here, if buying x is the best option for the user, then an honest advisor is more likely to assign 5 stars than 1 star to x . Some participants may have a hidden agenda, for example, to boost sales of product y . These participants provide ratings strategically, and we call them attackers.

The example shows that the actual details of the decision are not very important. The relationship between the abstract honest ratings and the decision is crucial for users. In this paper, we assert an arbitrary non-independent relationship between honest ratings and the decision. We do not concretely model this relationship (contrary to e.g. [11]).

4.1 Formal Model

A user makes a decision by picking one option from the ratings. And he tries to select the best option. We simply model the decision as a random variable, with its (unknown) outcome representing the (unknown) best option. The outcomes of a *decision* consists of a set of *options* $\mathcal{O} = \{0, \dots, n-1\}$. We use the random variable Θ over the options \mathcal{O} to denote the best option. Thus, $P(\Theta = \theta|\phi)$ is the probability that θ is the best option, when ϕ is given.

A rating has a certain format, it could be a number of stars (i.e. discrete and ordered), a list of tags (i.e. discrete and not ordered) or a real value in some range. On an abstract level, the structure is actually not that relevant. Specifically, it is only relevant when constructing an aggregation mechanism – which is not the purpose of this paper. We consider a *rating format* to be a set of scores \mathcal{R} , and a *rating* to be a random variable R , which has the property that it says something about Θ when the advisor is honest. To accommodate for multiple advisors giving ratings, let $\mathcal{A} = 0, \dots, m-1$ be the set of advisors, and let R_j be the rating provided by $j \in \mathcal{A}$. We use \mathbf{R}_A to mean R_{a_0}, \dots, R_{a_k} for $\{a_0, \dots, a_k\} = A \subseteq \mathcal{A}$.

Advisors can be honest or malicious. We introduce the status of advisor a , which is honest (\top) or malicious (\perp), as a random variable S_a . An honest advisor would typically not give the same rating as a malicious advisor. We introduce \widehat{R} and \widetilde{R} , both over \mathcal{R} , to represent the rating an honest or malicious advisor would give. Thus, $R_a = \widehat{R}_a$ whenever $S_a = \top$, and $R_a = \widetilde{R}_a$ whenever $S_a = \perp$. We shorthand the prior probability that a is honest as $P(S_a) = s_a$. As we reason about a trust system, we assert $0 < s_a < 1$.

In the running example, we mentioned that the honest advisors' ratings say something about the best option (decision). We distil that notion by saying Θ is not independent from honest advisors' ratings $A \subseteq \mathcal{A} : P(\Theta) \neq P(\Theta|\widehat{\mathbf{R}}_A)$. Another way to phrase this, is to say that honest advisors' ratings *leak* information about the decision. We need to use information theory to encode this notion [8]:

Definition 1. Let X, Y, Z be discrete random variables.

The surprisal of an outcome x of X is $-\log(P(x))$.

The entropy of X is

$$H(X) = \mathbb{E}_X(-\log(P(x))) = \sum_i P(x_i) \cdot -\log(P(x_i))$$

The conditional entropy of X given Y is

$$H(X|Y) = \mathbb{E}_X(-\log(P(x|y))) = \sum_{i,j} P(x_i, y_j) \cdot -\log(P(x_i|y_j))$$

The mutual information of X and Y is

$$I(X; Y) = \mathbb{E}_{X,Y}(\log(\frac{P(x,y)}{P(x)P(y)})) = \sum_{i,j} P(x,y) \log(\frac{P(x,y)}{P(x)P(y)})$$

The conditional mutual information of X and Y given Z is

$$I(X; Y|Z) = \mathbb{E}_Z(I(X; Y)|Z) = \sum_{i,j,k} P(x,y,z) \log(\frac{P(x,y|z)}{P(x|z)P(y|z)})$$

Information leakage of Y about X (given Z) is the (conditional) mutual information of X and Y (given Z). Information leakage is the difference in the information about X when Y is given and not given: $I(X; Y) = H(X) - H(X|Y)$, or $I(X; Y|Z) = H(X|Z) - H(X|Y, Z)$ [1]. Information leakage is non-negative.

Information theory allows us to rewrite the link between honest ratings and correct options as conditional information leakage:

$$\begin{aligned} & I(\Theta; \mathbf{R}_A | \mathbf{S}_A = \top) \\ &= H(\Theta | \mathbf{S}_A = \top) - H(\Theta | \mathbf{R}_A, \mathbf{S}_A = \top) \\ &= \sum_{\theta} p(\theta | \mathbf{S}_A = \top) - \sum_{\mathbf{r}_A} p(\mathbf{r}_A) \sum_{\theta} p(\theta | \mathbf{r}_A, \mathbf{S}_A = \top) \end{aligned}$$

We assume honest ratings are always correlated with correct options, hence there is always conditional information leakage: $I(\Theta; \mathbf{R}_A | \mathbf{S}_A = \top) > 0$.

Now, the dishonest advisors select R_j such that $I(\Theta; R_A | \Phi)$ is minimised for given Φ (such as $\Phi = \mathbf{S}_A$). The term, $I(\Theta; R_A | \Phi)$, quantifies how much the ratings say about the optimal option.

$$\begin{aligned} & I(\Theta; \mathbf{R}_A | \Phi) \\ &= H(\Theta | \Phi) - H(\Theta | \mathbf{R}_A, \Phi) \\ &= \sum_{\theta} p(\theta | \phi) - \sum_{\mathbf{r}_A, \phi} p(\mathbf{r}_A, \phi) \sum_{\theta} p(\theta | \mathbf{r}_A, \phi) \end{aligned}$$

We can use Example 1 to showcase parts of the formalisation:

Example 2. In Example 1, option “buy x ” becomes 0, “buy y ” 1 and “nothing” 2. The decision outcomes are $\{0, 1, 2\}$. We have 4 advisors, $\{0, 1, 2, 3\}$, and 0, 1 are suspicious with $P(S_0) = 0.2, P(S_1) = 0.3$, and 2, 3 are not with $P(S_2) = 0.8, P(S_3) = 0.9$. Ratings can be any of $\{(r, s) | r, s \in \{\emptyset, 1, 2, 3, 4, 5\}\}$.

4.2 Controlled Parameters

In the case of a centralised system, the designer himself can make these decisions. For decentralised systems, it may be users themselves to make decisions. In the latter case, the designer of the system should try to encourage users to make the right decisions. Either way, it is important to have theoretically rigorous guidelines to make robust decisions. Here, we look at the parameters of a system that can or cannot be controlled.

In this paper, we take the viewpoint of any party that wants to increase the robustness of the system. We refer to the parties that want to increase the minimal information leakage as the *defender*. For example, when we say “under the defender’s control”, we mean that the user, the designer or any party that strives for robustness controls it.

The set of advisors \mathcal{A} is not under the defender’s control. Moreover, for any advisor $a \in \mathcal{A}$, the random variables $S_a, \widehat{R}_a, \widetilde{R}_a$ and R_a cannot be controlled. However, the defender can blacklist/whitelist a subset of the advisors. Formally, the defender can choose $A \subseteq \mathcal{A}$ in $I(\Theta; \mathbf{R}_A)$. Moreover, in some systems, the defender can monitor which advisors potentially have information (e.g. which advisors have performed relevant transactions). If random variable K_a captures this fact for advisor a , then the defender may choose to have \mathbf{K}_A as a condition: $I(\Theta; \mathbf{R}_A | \mathbf{K}_A)$. Finally, the advisor may be able to change the actual decision, thus changing (the size of) the random variable Θ .

5 Limited Information Implies Inaccuracy

For the conclusions in this paper to hold relevance, we need to show that limited information leakage leads to limited accuracy. We do so constructively. In other words, we construct the best possible opinions that can result from given information leakage and some aggregation mechanism, and show that the accuracy of these opinions is limited by the amount of information leakage.

An opinion is an assignment of probability to each of the options in a decision. An opinion is said to be accurate, when the probability assigned to the right option is high. One way of measuring this is to take the cross entropy:

Definition 2. For discrete random variables X, Y with the same support, the cross entropy is

$$H_{cross}(X, Y) = \mathbb{E}_{x_i}(\log(P(y_i))) = - \sum_i P(x_i) \log(P(y_i))$$

The Kullback-Leibler divergence is

$$D_{KL}(X||Y) = H_{cross}(X, Y) - H(X) = \sum_i P(x_i) \log \frac{P(x_i)}{P(y_i)}$$

The cross entropy (and Kullback-Leibler divergence) is a standard tool to measure the quality of an approximation Y of the true distribution X . Specifically,

the cross entropy takes the expectation of the surprisal one has under the approximation Y . An advantage of Kullback-Leibler divergence is that the term $-H(X)$ translates the values, such that 0 divergence occurs when $X \sim Y$. Moreover, Kullback-Leibler divergence must be non-negative.

We use cross-entropy to measure accuracy. Let $O : \Theta \rightarrow [0, 1]$ such that $\sum_{\theta \in \Theta} O(\theta) = 1$ be an opinion. The accuracy of an opinion O is $-\sum_i P(\Theta = i) \log(O(i))$. The accuracy of O is limited by the information leakage of Θ . Specifically, given ratings \mathbf{R} , no matter how we select O , its accuracy cannot exceed a certain value, namely $H(\Theta|\mathbf{R})$. The theorem must state that O 's accuracy cannot exceed a threshold determined by the information leakage.

Theorem 1. *There is no opinion O , such that $-\sum_i P(\Theta = i|\mathbf{R}) \log(O(i))$ exceeds the threshold $H(\Theta) - I(\Theta; \mathbf{R})$.*

Proof. Using only standard notions from information theory: First, note $H(\Theta) - I(\Theta; \mathbf{R}) = H(\Theta|\mathbf{R})$. Second, $-\sum_i P(\Theta = i|\mathbf{R}) \log(O(i)) = H(\Theta|\mathbf{R}) - D_{KL}(\Theta||O)$, which suffices, since $D_{KL}(\Theta||O) \geq 0$. \square

5.1 Minimising Information Leakage

By definition, when all users are honest, there is non-zero information leakage. After all, if all users are honest $I(\Theta; \mathbf{R}_A) = I(\Theta; \widehat{\mathbf{R}}_A) > 0$. However, if some users are malicious, then there may not be information leakage. Formally:

Proposition 1. *There exist $A, \Theta, \widehat{R}, \widetilde{R}, S$, such that $I(\Theta; \mathbf{R}_A) = 0$.*

Proof. Take $A = \{a, b\}$, $P(\Theta=0) = 1/2 = P(\Theta = 1)$, $P(\widehat{R}=\Theta) = 1$, $P(\widetilde{R}=1 - \Theta) = 1$ and $P(S_a = h) = P(S_b = h) = 1/2$. Obviously, honest ratings leak (1 bit of) information, however, the actual ratings leak no information (about Θ). \square

On the other hand, it is not guaranteed for all A, Θ, \widehat{R} and S , a malicious strategy exists that achieves zero information leakage:

Proposition 2. *There exist $A, \Theta, \widehat{R}, S$, such that for all \widetilde{R} , $H(\Theta) - H(\Theta|\mathbf{R}_A) > 0$.*

Proof. Take A, Θ, \widehat{R} as in Proposition 1, but $P(S_a=h) = P(S_b=h) = 0.51$. Now $P(\Theta = 1|R_a = 1, R_b = 1) \geq P(\Theta = 1, S_a = h, S_b = h|R_a = 1, R_b = 1) \geq 0.51$. \square

Under certain specific circumstances, it is even possible to deduce exactly when it is possible for malicious advisors to block information leakage. The quantities depend on the exact assumptions. In [11–13], we looked at cases where the ratings perfectly match the option (i.e. full information leakage for honest users). For example, if malicious advisors are static and independent, the average probability of honesty must be below $1/n$, for n options [11]. In this paper, we do not quantify values, but study their relationships.

It may be possible for attackers to block information leakage (Proposition 1), but it may also be impossible (Proposition 2). Does the latter imply that there is no harmful attack? To answer that, we must determine the existence of an attack, such that the information leakage with the attack is lower than without. In fact, such an attack must always exist, provided that there is at least one user that has non-zero probability of being malicious.

Theorem 2. *For all $A, \Theta, \widehat{R}, S$, there exists \widetilde{R} such that $I(\Theta; \mathbf{R}_A) < I(\Theta; \widehat{\mathbf{R}}_A)$.*

Proof. Since $I(\Theta; \widehat{\mathbf{R}}_A)$, Θ and $\widehat{\mathbf{R}}_A$ are not independent, and there exists $\theta, \widehat{\mathbf{r}}_A$, such that $P(\theta|\widehat{r}) > P(\theta) + \epsilon$ and $P(\theta|\widehat{r}') < P(\theta) - \epsilon$ for some other rating \widehat{r}' . Take $P(\widetilde{R} = \widehat{r}|\theta) = P(\widehat{R} = \widehat{r}|\theta) - \epsilon$, and $P(\widetilde{R} = \widehat{r}'|\theta) = P(\widehat{R} = \widehat{r}'|\theta) + \epsilon$. All summands but two remain the same: $P(\theta, \widehat{r}) \log P(\theta|\widehat{r}) + P(\theta, \widehat{r}') \log P(\theta|\widehat{r}')$ are closer to their average, we can apply Jensen's inequality to get the theorem. \square

So far, we have proven that some attacks may block all information leakage, but that such an attack may not exist, and that, nevertheless, a harmful attack must exist, except in trivial cases. These results suggest the possibility that all attacks reduce information leakage. However, this is not the case. There exist attacks that increase the information leakage:

Proposition 3. *There exist $A, \Theta, \widehat{R}, \widetilde{R}, S$, such that $I(\Theta; \mathbf{R}_A) > I(\Theta; \widehat{\mathbf{R}}_A)$.*

Proof. Take A and Θ as in Proposition 1. Take $P(\widehat{R} = \Theta) = 0.6$, $P(\widetilde{R} = \Theta) = 0.7$ and $0 < P(\mathbf{S}_A) < 1$. The inequality is satisfied with these values. \square

Notice that in the proof of Proposition 3, the information leakage of \widetilde{R} is (strictly) greater than \widehat{R} . This is a necessary condition for an attack not to be harmful. However, it is not a sufficient condition. If we had taken $P(\widetilde{R} = \Theta) = 0.3$, then the information leakage of \widetilde{R} remains the same, but the information leakage of R decreases (as long as $P(S)$ is not close to 0).

A realistic scenario where Proposition 3 could apply, is a camouflage attack. In the camouflage attack, an advisor provides high quality ratings (i.e. high information leakage), to gain trust, and later abuses the trust for a specific goal. In [13], we have studied these camouflage attacks, and identified that an existing attack is actually not harmful. Furthermore, we found that a probabilistic version of the camouflage attack can minimise information leakage.

If we want a trust system to be robust, then it must be able to deal graciously with the all malicious ratings, including the ones that minimise information leakage. The ratings that minimise information leakage are referred to as the *minimal* $\widetilde{\mathbf{R}}_A$. Which ratings minimise information leakage depends on $A, \Theta, \widehat{\mathbf{R}}_A$ and \mathbf{S}_A .

6 Design Guidelines

This section is the core of the paper. Here, we study choices that one can make to mitigate unfair rating attacks. This section is divided into subsections, each of which considers an individual choice. To give a quick overview of the results:

1. It is not helpful to block seemingly malicious advisors, and often counterproductive.
2. When the number of advisors is limited, seemingly honest advisors should be preferred.
3. Disregard advisors that should not have information about the decision; e.g. buyers that never bought from a seller.
4. When forced to choose between two seemingly equally honest advisors, the better-known advisor should be preferred.
5. Different groups of honest advisors whose ratings may have the same information leakage, but different robustness towards attackers. We find a property that characterises the robustness of ratings from equally informative groups of honest advisors. This shows that for a reasonable way to increase the size of a decision, information leakage increases.

In the relevant sections, we not merely show the results, but, more importantly, we analyse and interpret them. Some our suggestions are already widely adopted. However, they are adopted for reasons other than robustness. Moreover, our guidelines are based on a solid information-theoretic foundation.

6.1 Blocking Malicious Advisors

Theorem 2 states that the minimum information leakage of ratings is strictly smaller than the honest ratings. So problematic attacks reduce information leakage. Perhaps, we can robustly increase the information leakage, by blocking suspicious advisors. On the other hand, blocking suspicious advisors may decrease the information leakage, as even a suspicious advisor may provide honest ratings. We show in this section, that the latter holds: Blocking malicious advisors does not increase the robustness against unfair rating attacks.

First, we start with a weak version of the theorem, that directly refutes the intuition that sufficiently suspicious advisors must be blocked. We introduce a threshold of suspicion c , such that only those ratings from advisors at or above the threshold are considered. If indeed it helps to block sufficiently suspicious advisors, then such a c must exist. However, this is not the case:

Proposition 4. *For all A , Θ , \widehat{R} , \widetilde{R} and S , there is no threshold $c \in [0, 1]$, with $A^{\geq c} \subseteq A$ as the set of advisors such that $s_a \geq c$, such that $I(\Theta; \mathbf{R}_{A^{\geq c}}) > I(\Theta; \mathbf{R}_A)$.*

Proof. Since $H(X|Y, Z) \leq H(X|Y)$, $-H(\Theta|\mathbf{R}_{A^{\geq c}}) \leq -H(\Theta|\mathbf{R}_A)$. □

For a pair $c < d$, we can let $A' = A^{\geq d}$ and automatically $A'^{\geq c} = A^{\geq c}$, and the proposition applied to A' proves that $I(\Theta; \mathbf{R}_{A^{\geq c}}) > I(\Theta; \mathbf{R}_A^{\geq d})$. Therefore, Proposition 4 proves *monotonicity* of $I(\Theta; \mathbf{R}_{A^{\geq c}})$ over c .

For the vast majority of thresholds, however, blocking suspicious advisors is not just ineffective, but actually harmful. Thus, for some (small but non-zero) blocking thresholds, blocking does not alter information leakage, but for most thresholds – including all thresholds over $1/2$ – blocking strictly decreases information leakage:

Theorem 3. *For all $A, \Theta, \widehat{R}, S$ and minimal \widetilde{R} , there exists a threshold $d \in [1/n, 1)$ such that $I(\Theta; \mathbf{R}_{A^{\geq c}}) = I(\Theta; \mathbf{R}_A)$ iff $c \leq d$.*

Proof. Note that if $c = 0$ then trivially the equality holds, and if $c = 1$, then the equality trivially does not hold, since $s_a < 1$, $A^{\geq c} = \emptyset$ and thus $I(\Theta; \mathbf{R}_{A^{\geq c}}) = 0 < I(\Theta; \mathbf{R}_A)$. Using the monotonicity proved in Proposition 4, it suffices to prove the equality for $c = \frac{1}{n}$: Now $P(R_b|\Theta) = s_a P(\widehat{R}_b|\Theta) + (1 - s_a) P(\widetilde{R}_b|\Theta)$, and if $s_a \leq \frac{1}{n}$, there exists \widetilde{R}_b such that $P(R_b|\Theta) = \frac{1}{n}$, meaning the minimum \widetilde{R}_b can achieve zero information leakage for $c \leq \frac{1}{n}$. \square

Arguably, Proposition 4 is not particularly interesting from an information-theoretic perspective – although it may be somewhat surprising superficially. After all, meaningless additional information does not decrease the information leakage. However, Theorem 3 strengthens the result to say that there exists a level of suspicion below which blocking is harming robustness. In [11, 13], we show that for typical systems, this threshold is very low.

Design Hint 1. *Unless the suspicion that an advisor is malicious is extremely high, blocking a suspicious advisor is either useless or counterproductive for robustness.*

6.2 More Honest Advisors

The reason that blocking advisors does not help is the simple theorem that more random variables in the condition cannot decrease information leakage. However, clearly, a seemingly honest advisor contributes more information than a suspicious one. There may be a cost associated to requesting/receiving too many ratings, or other reasons why the number of advisors is limited. In these cases, we expect that preferring those advisors that are more likely to be honest is better for gaining information.

Take two near-identical trust systems, that only differ in the degree of honesty of the advisors. The decisions and honest ratings remain the same, and the attacker minimises information leakage on both sides. Then, the more honest system has higher information leakage:

Theorem 4. *For all $A, A', \Theta, \widehat{R}, S$, such that $|A| = |A'|$ and $P(S_{a_i}) \geq P(S_{a'_i})$, if \widetilde{R} and \widetilde{R}' are minimising, then $I(\Theta; \mathbf{R}_A) \geq I(\Theta; \mathbf{R}_{A'})$.*

Proof. The attacker can select $\tilde{\mathbf{R}}_A'$ such that $P(\mathbf{R}_A = x|\varphi) = P(\mathbf{R}_{A'} = x|\varphi)$, for any condition φ . The minimum information leakage is at most equal to the construction's information leakage. \square

Design Hint 2. *When there is a cost to gathering too many ratings, then seemingly more honest advisors should be asked before more suspicious ones.*

6.3 Unknowing Advisors

The first question is, whether it is useful to be aware of whether advisors could have knowledge. We introduce a random variable K_a , such that $K_a = 0$ if a does not have knowledge, and $K_a = 1$ if a may have some knowledge. Formally, $I(\Theta; \tilde{R}_a | K_a = 0) = 0$ and $I(\Theta; \tilde{R}_a | K_a = 1) > 0$. Now we can reason about the difference in information with and without K_a as a condition: $I(\Theta; \mathbf{R}_A)$ is the information leakage without knowing whether advisors could have knowledge, and $I(\Theta; \mathbf{R}_a | \mathbf{K})$ is the expected information leakage when we know whether advisors could have knowledge. In fact, in the latter case, we have at least as much information leakage:

Theorem 5. *For all A, Θ, \mathbf{R}_A and \mathbf{K} , $I(\Theta; \mathbf{R}_A) \leq I(\Theta; \mathbf{R}_A | \mathbf{K})$*

Proof. Since $H(\Theta | \mathbf{K}) = H(\Theta)$, $I(\Theta; \mathbf{R}_A | \mathbf{K}) = I(\Theta; \mathbf{R}_a, \mathbf{K})$. And additional conditions do not increase entropy, hence $I(\Theta; \mathbf{R}_A, \mathbf{K}) \geq I(\Theta; \mathbf{R}_a)$. \square

Based on that result, we can revisit the notion of blocking users. Should we block unknowing advisors? It turns out that blocking unknowing advisors never changes information leakage:

Corollary 1. *For arbitrary conditions ψ, φ , and $a \in \mathcal{A}, \Theta, R_a, K_a$, we have $I(\Theta; R_a, \psi | K_a = 0, \varphi) = I(\Theta; \psi | \varphi)$*

Proof. Since $I(\Theta; \hat{R}_a | K_a = 0, \varphi, \psi) = 0$, the optimal strategy for the malicious advisor is to set \tilde{R}_a to satisfy $I(\Theta; \tilde{R}_a | K_a = 0, \varphi, \psi) = 0$, which implies $I(\Theta; R_a | K_a = 0, \varphi, \psi) = 0$. Then, R_a can be eliminated without loss of generality, and then also K_a . \square

Design Hint 3. *When possible, keep track of whether an advisor can provide useful advise; e.g. because he actually performed a transaction. Advisors that cannot provide useful advise can be filtered out without loss of generality.*

6.4 Newer Advisors

Throughout the paper, we have ignored previous ratings. However, as we show in [13], if we learn about the correctness of an advisor's rating in hindsight, then we learn about the status of that advisor. Concretely, for some random variable Q , if $P(Q = 1 | R_a = \hat{R}_a) > P(Q = 1 | R_a = \tilde{R}_a)$, then $P(S_a | Q) \neq P(S_a)$. Here Q corresponds to the probability that an advisor's rating is correct in hindsight.

To keep the notation simple, we did not introduce Q previously. Here, we need it to prove a theorem.

For an honest user, R_a is always equal to \widehat{R}_a , and an attacker may or may not set R_a equal to \widehat{R}_a (by selecting $\widehat{R}_a = \widetilde{R}_a$). Often, there may be multiple Q 's per advisor, so we use $\mathbf{Q}_a^{\leq i}$ to denote Q_1, Q_2, \dots, Q_i . Let a, a' be advisors, such that $P(S_a | \mathbf{Q}_a^{\leq i}) = P(S_{a'} | \mathbf{Q}_{a'}^{\leq j})$ and $i < j$. Then, a and a' are equally suspicious (their statuses are equiprobable with the given conditions), but a is a newer advisor than a' . For the current rating, a and a' are equally useful (as they are equally likely to be honest), but if the current rating will be correct in hindsight, then a loses suspicion quicker than a' .

For example, let a be completely new – its suspicion is the prior $P(S_a)$ – and a' be older, but have both seemingly correct and seemingly wrong ratings in hindsight, such that $P(S_{a'} | \mathbf{Q}) = P(S_a)$. Clearly, if we got a seemingly correct rating from a in hindsight, its suspicion level changes much more radically than when the rating of a' was correct in hindsight. The same holds if the rating was seemingly wrong. However, if we only select a or a' , then the other advisor's suspicion level remains unchanged. Therefore, if we must select either a or a' and both are equally suspicious, then we should prefer the newer advisor.

To formalise this concept, measure the information leakage over two ratings, where we switch advisor if their ratings seemed wrong in hindsight:

Theorem 6. *For a, a' , such that $P(S_a | \mathbf{Q}_a^{\leq i}) = P(S_{a'} | \mathbf{Q}_{a'}^{\leq j})$ and $i < j$:*

$$I(\Theta; R_a | \mathbf{Q}_a^{\leq i}) + P(Q=0 | R_a, \Theta) \cdot I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j}) + P(Q=1 | R_a, \Theta) \cdot I(\Theta; R_a | \mathbf{Q}_a^{\leq i}, Q=1) \geq I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j}) + P(Q=0 | R_a, \Theta) \cdot I(\Theta; R_a | \mathbf{Q}_a^{\leq i}) + P(Q=1 | R_a, \Theta) \cdot I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j}, Q=1).$$

Proof. Note that since $I(\Theta; R_a | \mathbf{Q}_a^{\leq i}) = I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j})$, all terms on both sides are equal, except $I(\Theta; R_a | \mathbf{Q}_a^{\leq i}, Q)$ and $I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j}, Q)$. Hence it suffices to prove $I(\Theta; R_a | \mathbf{Q}_a^{\leq i}, Q=1) \geq I(\Theta; R_{a'} | \mathbf{Q}_{a'}^{\leq j}, Q=1)$.

The Q 's are independent of Θ , meaning that it suffices to prove $H(\Theta | R_a, \mathbf{Q}_a^{\leq i}, Q=1) \leq H(\Theta | R_{a'}, \mathbf{Q}_{a'}^{\leq j}, Q=1)$. Now, $P(\Theta | R_a, \mathbf{Q}_a^{\leq i}, Q=1) = P(\Theta | R_a, S_a^*)$, with $S_a^* = S_a | \mathbf{Q}_a^{\leq i}, Q=1$, and similarly on the other side. Thus, if $\mathbb{E}(S_a^*) \geq \mathbb{E}(S_{a'}^*)$, then the theorem follows. Since the Q 's are Bayesian updates, we can model S^* as a Beta distribution times an arbitrary prior. The expectation of the Beta distribution is $\frac{p+1}{p+n+2}$, which is more sensitive to increasing p when $p+n$ is small. \square

Design Hint 4. *When two advisors appear equally likely to be honest, but only one may provide a rating, then the advisor with whom we have the shortest history should be preferred.*

6.5 More Options

The defender may want to increase the robustness of the rating system, by changing the rating format or the number of choices in a decision. There is an immediate problem when formalising this idea, which is that we have not formalised how honest users respond when either the domain of Θ or R changes.

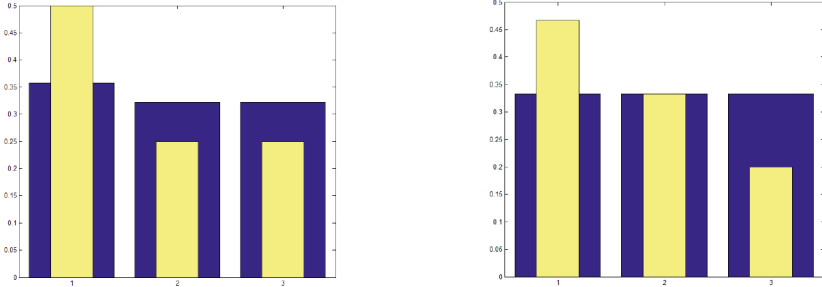


Fig. 1. Two ratings (blue), the corresponding honest ratings (yellow), with the values of θ on the x-axis, and the corresponding probability on the y-axis. (Color figure online)

Naively, we may simply assert that the information leakage of honest ratings remains unchanged. Thus $I(\Theta; \widehat{\mathbf{R}}_A) = I(\Theta'; \widehat{\mathbf{R}}'_A)$. However, the robustness of the system on the left is not equal to that on the right.

Theorem 7. For given $A, \Theta, \Theta', \widehat{\mathbf{R}}, \widehat{\mathbf{R}}'$ with minimising $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{R}}'$, $I(\Theta; \widehat{\mathbf{R}}_A) = I(\Theta'; \widehat{\mathbf{R}}'_A)$ does not imply $I(\Theta; \mathbf{R}_A) = I(\Theta'; \mathbf{R}'_A)$.

Proof. Let the prior $P(\Theta) = P(\Theta') = 1/3$. Moreover let $P(\Theta = i | \widehat{\mathbf{R}}_A = i) = 1/2$ and $P(\Theta = i | \widehat{\mathbf{R}}_A \neq i) = 1/4$; meaning $I(\Theta; \widehat{\mathbf{R}}_A) = 1.5$. Then, we let $P(\Theta' = i | \widehat{\mathbf{R}}'_A = i) = 7/15$, then it follows from the fact that the information leakage is 1.5 that $P(\Theta' = i | \widehat{\mathbf{R}}'_A \equiv_3 i + 1) \approx 0.195 \approx 1/5$ and $P(\Theta' = i | \widehat{\mathbf{R}}'_A \equiv_3 i + 2) \approx 0.338 \approx 1/3$ (or vice versa). See Fig. 1. When $P(\mathbf{S}_A) \approx 5/7$, we can achieve 0 information leakage by setting $P(\Theta' = i | \widehat{\mathbf{R}}'_A \equiv_3 i + 1) = 1/3$, and $P(\Theta' = i | \widehat{\mathbf{R}}'_A \equiv_3 i + 1) = 2/3$, then it follows that $P(\Theta' | \mathbf{R}'_A) = 1/3 = P(\Theta' = i)$. Thus, for $P(\mathbf{S}_A) \approx 5/7$ we have 0 information leakage for Θ' , but since $P(\Theta = i | \mathbf{R}_A = i) \geq P(\Theta = i | \widehat{\mathbf{R}}_A = i) \cdot 5/7 = 5/14$, $P(\Theta = i | \mathbf{R}_A = i) \neq 1/3 = P(\Theta = i)$, there is non-zero information leakage for Θ . Hence their robustness is different. \square

The proof is visualised in Fig. 1, where the yellow/light bars are the honest ratings, and the blue/dark bars are the overall ratings. The honest ratings have the same information leakage in both graphs, whereas the overall ratings clearly do not.

More choices generally mean more information leakage. However, as proven in Theorem 7, special circumstances must be met when expanding Θ . In particular, Θ is expanded together with R in an orthogonal way – the additional options have their own corresponding ratings:

Theorem 8. Given two rating systems that one has more options for ratings and choices: $|\Theta| = |\Theta'| + 1, |\mathbf{R}_A| = |\mathbf{R}'_A| + 1$, if $p(\theta|r) = p(\theta'|r')$, then $I(\Theta; R_a)$ can be equal, or larger than $I(\Theta'; R'_a)$

Proof. Let $p(\theta|r) = 0$ for either $\theta > |\Theta'|$ or $r > |\mathbf{R}'_A|$, except that there is a $\dot{r} > |\mathbf{R}'_A|$ and a $\dot{\theta} > |\Theta'|$, for which $p(\dot{\theta}|\dot{r}) = 1$. We get $H(\Theta) = H(\Theta') - \mathbf{f}(p(\dot{r}))$, and $H(\Theta|R_a) = H(\Theta'|R'_a) - p(\dot{r}) \cdot \mathbf{f}(p(\dot{\theta}|\dot{r})) = H(\Theta|R'_a)$. Given $p(\dot{r}) \geq 0$, $H(\Theta) - H(\Theta|R_a) \geq H(\Theta') - H(\Theta'|R'_a)$. Hence, $I(\Theta; R_a) \geq I(\Theta'; R'_a)$. \square

Intuitively, one of two ratings can happen, if the additional rating occurs, the user gains a lot of information – namely that the additional option occurred. The remaining cases do not involve the new rating or choice, and remains essentially unchanged (only linear weights change proportionally).

Design Hint 5. *Increasing the number of options in a decision is good, assuming that the additional options are sufficiently distinctive for the advisors. Care should be taken, because additional options may harm robustness in some cases.*

7 Conclusion

Users form opinions based on ratings. Systems that allow more accurate opinions are better. We use cross entropy (or Kullback-Leibler divergence) to measure the accuracy of opinions. The maximum accuracy is limited by a quantity called information leakage. Information leakage measures how much a rating tells about the decision a user wants to make.

The amount of information leakage of honest ratings has a certain non-zero quantity. Thus, we assume that there is some correlation between honest ratings and what the best decision is. We cannot make such assumptions about ratings from attackers. Attackers have a hidden agenda that they base their ratings upon. We want to reduce the negative effect that attackers have on the information leakage. To be on the safe side, we assume that attackers rate in the way that minimises the information leakage – any other behaviour results in at least as much information leakage.

Our model of a rating system is abstract. Decisions and ratings are abstract entities. Ratings are not under the control of the defender. However, the defender can select which advisors to use, potentially monitor whether an advisor may be knowledgeable, and consider more options in his decision. Our main contribution is a set of guidelines for the defender to use these factors in his advantage.

Our main guidelines are: Blocking suspicious advisors can only decrease robustness (1). If only a limited number of ratings can be used, however, then less suspicious advisors are better (2), and in case of a tie, newer advisors are better (3). Observing transactions increases robustness (4). Offering more choices may increase robustness (5).

References

1. Cover, T.M., Thomas, J.A.: Entropy, relative entropy and mutual information. *Elem. Inf. Theory*, 12–49 (1991)
2. Fang, H., Bao, Y., Zhang, J.: Misleading opinions provided by advisors: dishonesty or subjectivity. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1983–1989 (2013)

3. Nan, H., Liu, L., Sambamurthy, V.: Fraud detection in online consumer reviews. *Decis. Support Syst.* **50**(3), 614–626 (2011)
4. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **43**(2), 618–644 (2007)
5. Jurca, R., Faltings, B.: Minimum payments that reward honest reputation feedback. In: *Proceedings of the 7th ACM Conference on Electronic Commerce*, pp. 190–199. ACM (2006)
6. Jurca, R., Faltings, B.: Collusion-resistant, incentive-compatible feedback payments. In: *Proceedings of the 8th ACM Conference on Electronic Commerce*, pp. 200–209. ACM (2007)
7. Kerr, R., Cohen, R.: Smart cheaters do prosper: defeating trust and reputation systems. In: *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 993–1000. IFAAMAS (2009)
8. Robert, J.: *McEliece: Theory of Information and Coding*, 2nd edn. Cambridge University Press, New York (2001)
9. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 1206–1212 (2006)
10. Luke Teacy, W.T., Luck, M., Rogers, A., Jennings, N.R.: An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artif. Intell.* **193**, 149–185 (2012)
11. Wang, D., Muller, T., Irissappane, A.A., Zhang, J., Liu, Y.: Using information theory to improve the robustness of trust systems. In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 791–799 (2015)
12. Wang, D., Muller, T., Zhang, J., Liu, Y.: Quantifying robustness of trust systems against collusive unfair rating attacks using information theory. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 111–117 (2015)
13. Wang, D., Muller, T., Zhang, J., Liu, Y.: Is it harmful when advisors only pretend to be honest? In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (2016)
14. Weng, J., Shen, Z., Miao, C., Goh, A., Leung, C.: Credibility: how agents can handle unfair third-party testimonies in computational trust models. *IEEE Trans. Knowl. Data Eng. (TKDE)* **22**(9), 1286–1298 (2010)
15. Whitby, A., Jøsang, A., Indulska, J.: Filtering out unfair ratings in bayesian reputation systems. In: *Proceedings of the AAMAS Workshop on Trust in Agent Societies (TRUST)*, pp. 106–117 (2004)
16. Yang, Y., Sun, Y., Kay, S., Yang, Q.: Securing rating aggregation systems using statistical detectors and trust. *IEEE Trans. Inf. Forensics and Secur.* **4**(4), 883–898 (2009)
17. Han, Y., Shen, Z., Miao, C., An, B., Leung, C.: Filtering trust opinions through reinforcement learning. *Decis. Support Syst.* **66**, 102–113 (2014)
18. Zhang, J., Cohen, R.: Design of a mechanism for promoting honesty in e-marketplaces. In: *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, pp. 1495–1500 (2007)

Enhancing Business Process Models with Trustworthiness Requirements

Nazila Gol Mohammadi^(✉) and Maritta Heisel

paluno - The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Essen, Germany
{nazila.golmohammadi,maritta.heisel}@paluno.uni-due.de

Abstract. The trustworthiness of systems that support complex collaborative business processes is an emergent property. In order to address users' trust concerns, trustworthiness requirements of software systems must be elicited and satisfied. The aim of this paper is to address the gap that exists between end-users' trust concerns and the lack of implementation of proper trustworthiness requirements in software systems. We focus on the challenges of specifying trustworthiness requirements and integrating them into the software development process as business process models. This paper provides a conceptual model of our approach by extending Business Process Model and Notation (BPMN) for integrating trustworthiness requirements. Our proposed approach explicitly considers the trustworthiness of individual components as part of the business process models. We use an application example from the health care domain to demonstrate our approach.

Keywords: Trust · Trustworthiness · Requirements · Business process modeling

1 Introduction

Advances on Information and Communication Technology (ICT) facilitate the automation of business processes and consequently increase organizations' efficiency. However, using new ICTs like cloud computing can also bring undesirable side effects, e.g., introducing new vulnerabilities and threats caused by collaboration and data exchange over the Internet. The consumers of business processes (either organizations or individuals) often hesitate in placing their trust in such technologies. Since trust is the prerequisite for performing many kinds of transactions and collaborations, users' concerns about the trustworthiness of these business processes, their involved apps, systems and platforms, slow down their adoption [6].

Business process models are frequently used in software development for understanding the behavior of the users, their requirements and for the assignment of requirements to particular well-defined business process elements.

In business processes, resources are either human or non-human assets, e.g., software, apps or IT devices [3]. Non-human assets can provide either fully-automated or semi-automated support to the activity performers. Since people rely on these technical resources when performing their activities, trustworthiness properties of these technical resources play a major role in gaining the trust of end-users (e.g., the reliability of the system that deals with monitoring the vital signs of a patient). There are specific conditions that must be defined concerning human resources that contribute as well to trustworthiness, e.g., people's skills and expertise when performing particular tasks. In addition to trustworthiness requirements on resource management, the usage of digital documents and data plays a central role in the trustworthiness. For instance, in order to respect privacy regulations, digital documents have to be protected from unauthorized use (e.g., being shared in public networks). This clearly demands the consideration of trustworthiness properties, and hence the specification of trustworthiness requirements on data objects by defining usage rules, as well as the respective mechanisms for enforcing the usage of such rules. Consequently, trustworthiness should be considered in the management of both human and non-human resources in all stages of the business process life-cycle: design, modeling, implementation, execution, monitoring and analysis.

In the state of the art, issues related to security have been widely studied. Since trustworthiness covers a broader spectrum of properties rather than just security, there is a gap in research when addressing socio-economical factors of trustworthiness [10]. Especially software systems that provide support to different stakeholders should fulfill a variety of qualities and properties for being trustworthy, depending on application and domain [9]. For instance, organizations require confidence about their business-critical data, whereas an elderly person using a health care service may be more concerned about reliability and usability.

In this paper, we aim at closing the existing gap between end-users' trust concerns and the lack of implementation of the appropriate trustworthiness properties in software systems. We focus on specifying trustworthiness requirements starting from the business processes level by providing modeling capabilities to understand and express trustworthiness requirements. Our approach specifies which functionalities with which qualities should be realized to address trustworthiness and gain the trust of the end-user. For instance, one of the factors for gaining trust is awareness. Business processes should include transparency capabilities either in the form of functionalities or qualities, e.g., defining notification activities or escalation events upon activities on users sensitive data. Usability and quality of representation of this notification are quality-related aspects. We specify which kind of transactions and activities need to be transparent to which extent for which organization or users. We mainly contribute to (1) understanding trustworthiness requirements and integrating them into the business process model, and (2) delivering detailed documentation of trustworthiness requirements along with the business process models using Business Process Model and Notation (BPMN) [17].

Tools and services developers are supported through detailed trustworthiness requirements for the software and services to be built. Then, based on trustworthiness requirements embedded in business process models, they can make more informed design decisions. We also believe that once trustworthiness requirements have been considered and documented in business process models, they will not be ignored during design-time. To demonstrate the enhancement of business process models with trustworthiness requirements, we consider an example from the health care domain, namely, an Ambient Assisted Living (AAL) system.

The remainder of this paper is structured as follows: Sect. 2 provides a brief overview of the fundamental concepts and the background. Section 3 presents an overview of the state of the art. Section 4 describes the classification of trustworthiness requirements, which can be expressed in the business process model. Furthermore, it gives initial recommendations for modeling and documenting trustworthiness-related capabilities into the business process. Section 5 demonstrates our approach using an application example from AAL. Section 6 presents conclusions and future work.

2 Fundamental Concepts and Background

This section introduces the notion of trust and moves on to define the meaning of trustworthiness. We then identify the relation between trust and trustworthiness. The basis of this work has been built up on the definition of trust and trustworthiness in our previous works in [9,10]. We distinguish between these two concepts.

Trust and Trustworthiness. *Trust* is defined as a “bet” about the future contingent actions of a system [22]. The components of this definition are belief and commitment. There is a belief that placing trust in a software or a system will lead to a good outcome. Then, the user commits the placing of trust by taking an action by using the business process and its software systems. This means, when a user decides to use a service, e.g., a health care service on the web, then he/she is confident that it will meet his/her expectations. Trust is subjective and different from user to user. For instance, organizations require confidence about their business-critical data, whereas an elderly person using a health care service (end-users) may be more concerned about usability. These concerns manifest themselves as trustworthiness requirements. Thus, business processes and their involved software systems and services need to be made trustworthy to mitigate the risks in engaging those systems and trust concerns of their users.

Trustworthiness properties are qualities of the system that potentially influence trust in a positive way. The term *trustworthiness* is not used consistently in the literature. Trustworthiness has sometimes been used as a synonym for security and sometimes for dependability. However, security is not the only aspect of trustworthiness. Some approaches merely focus on single trustworthiness characteristics, e.g., security or privacy. Most existing approaches have assumed that one-dimensional properties of services lead to trustworthiness, and even to trust

in it by users, such as a certification, the presence of certain technologies, or the use of certain methodologies. However, trustworthiness is rather a broad-spectrum term with notions including reliability, security, performance, and usability as parts of trustworthiness properties [15]. Trustworthiness is domain and application dependent. For instance, in health care applications, the set of properties which have primarily been considered consists of availability, confidentiality, integrity, maintainability, reliability and safety, but also performance and timeliness. Trustworthiness depends on a specific context and goals [9].

For instance, in safety-critical domains the failure tolerance of a system might be prioritized higher than its usability. We, furthermore, need to consider different types of components, e.g., humans as social parts of the system or software assets as technical ones. Trustworthiness in general can be defined as the assurance that the system will perform as expected [9]. With a focus on business processes, we adopt the notion of trustworthiness from [9], which covers a variety of trustworthiness properties as contributing to trust. This allows us to consider trustworthiness as the degree to which relevant qualities (then referred to as trustworthiness properties) are satisfied.

Business Process Models. A business process model is the representation of the activities, documents, people and all the elements involved in a business process, as well as the execution constraints between them [4]. BPMN [17] is the standard for modeling business processes, which is extended and used widely in both, industry and research. Most important BPMN elements are as follows:

- Activities are depicted as rounded rectangular boxes.
- Events, which include receiving and triggering events, are depicted as circles.
- Data objects are depicted as a sheet of paper with the top right corner folded.
- Gateways, control of how the process flows, are depicted as diamonds.

An important feature of business process modeling is to create high-level, domain-specific models or abstractions rather than focus on platform-specific models which often involve details and dependencies of implementation and execution environments [12]. Business Process Management (BPM) is also considered to be a key driving force in building, maintaining, and evolving enterprise applications and an agile software development technology which transforms business strategies into IT executions in a fast and standardized way [2].

3 Related Work

The study of related work reveals some gaps in resource management in BPM with respect to trustworthiness. Several works have been performed to overcome the problem of resource assignment, some meta-models like [13, 25] and an expressive resource assignment language [3] have been developed. That language, RALPH [3], provides a graphical representation of the resource selection conditions and assignments. RALPH has a formal semantics, which makes it appropriate for automated resource analysis in business process models. Stepien et al. [20]

present the user interfaces in which users can define the conditions themselves. The main gap is to address the broad spectrum of qualities which contribute to trustworthiness, and the necessity of defining conditions on resources and activities in business processes with respect to trustworthiness.

Plenty of works are done on security and to some extent on privacy. Short et al. [19] provide an approach for dealing with the inclusion of internal and/or external services in a business process that contains data handling policies. Wang et al. [26] developed a method to govern adaptive distributed business processes at run-time with an aspect-oriented programming approach. Policies can be specified for run-time governance, such as safety constraints and how the process should react if they are violated.

Resource patterns [18] are used to support expressing criteria in resource allocations. Business Activities is a Role-based access control (RBAC) [21] extension of Unified Modeling Language (UML) activity diagrams to define the separation of duties and binding of duties between the activities of a process. Wolter et al. [27] developed a model-driven business process security requirement specification which introduces security annotations into business process models for expressing security requirements for tasks.

However, the current state of the art in this field neglects to consider trustworthiness as criteria for the resources and business process management.

4 Modeling Trustworthiness Requirements in Business Processes Level Using BPMN

Trustworthiness requirements are usually defined first on a technical level, rather than on a business process level. However, at the business process level, we are able to provide a comprehensive view on the participants, the assets/resources and their relationships regarding satisfaction of business goals, as well as trustworthiness goals. Integrating trustworthiness-related information into business processes will support designers and developers in making their design decisions. Trustworthiness requirements on the business process level can be translated into concrete trustworthy configurations for service-based systems. Therefore, our proposed approach can be applied on different abstraction levels. Figure 1 shows how trustworthiness requirements provided by our approach will streamline the software development. The left side of Fig. 1 shows the level of abstraction for trustworthiness and their influences on different levels of abstraction on the system-side (simplified SOA layers). The refinement of trustworthiness requirements on different abstraction levels with a combination of goal models and business process models is presented in our other work [7].

The method for systematic identification and analysis of trustworthiness requirements is shown in Fig. 2. Our proposed method uses goal and business process modeling, iteratively. Here, we only focus on enriching business process models with trustworthiness. The method starts with a context analysis. The major task of context analysis which we are interested in here is “*identification of end-user trust concerns*”. Prior to this step, the participants of a business

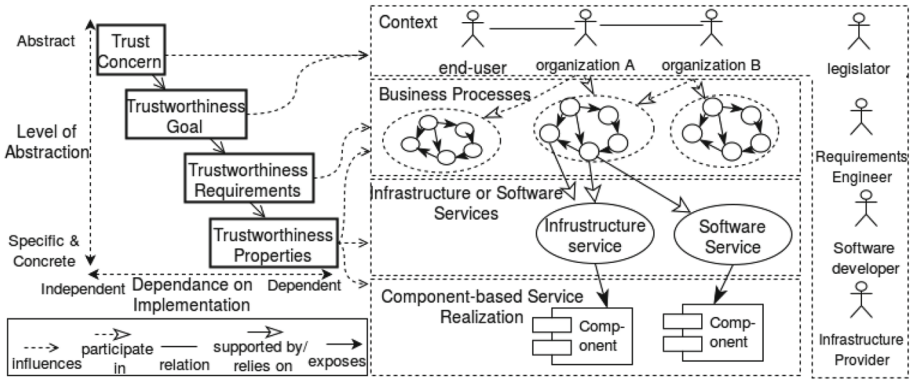


Fig. 1. Placing our proposed approach for enriching business processes with trustworthiness requirements and their alignment with software development

and stakeholders are captured. We assume this information about the context is provided in a context model. This step is concerned with providing a list of trust concerns for the end-users. These trust concerns are captured by interviewing end-users, based on expertise of a requirements engineer. We provide a questionnaire to support the requirements engineer by identification of end-users’ trust concerns [8]. Trust concerns and their dependencies on other participants in the business will be identified. Trust concerns are subjective and also domain and application dependent. The top-level business goals of identified stakeholders and business participants are captured in the goal models. We assume the goal models with the major intention of these involved parties/stakeholders are given. For satisfying the goals and presenting how they are realized, the business process models are set up. To support this step, a catalogue of trustworthiness attributes which mitigate trust concerns is provided in our previous work in [10]. Next, based on trust concerns we “*identify the trustworthiness goals*”. The initial goal model will be refined and updated with trustworthiness goals and its relation to the other goals. We *select one of the business process models* for including

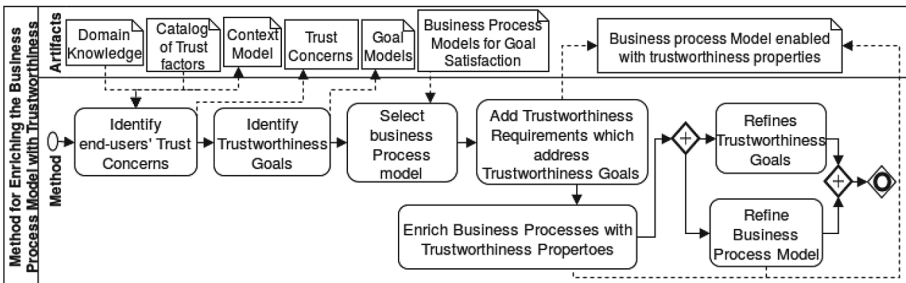


Fig. 2. Our method for enriching business processes with trustworthiness requirements

trustworthiness requirements satisfying trustworthiness goals. This selection is based-on the location of the trustworthiness goal to the other goals. This steps goes through business process elements and control flow and questions whether a specific element in the business process is trustworthiness-related. Refinement of the business process model details business processes with including more concrete trustworthiness properties on resources, activities, etc. for satisfying trustworthiness requirements. This step can be concurrent to the goal and trustworthiness goal refinements, and both models can iteratively develop. Figure 2 gives an overview of the above mentioned steps and their input and output artifacts.

In this work, we focus only on specifying trustworthiness requirements in business process models. We propose a BPMN extension that allows the integration of trustworthiness requirements into a business process. We introduce trustworthiness elements for business process modeling which allows modeling and documenting trustworthiness requirements as well as placing a control to address the trust concerns of the end-users. Later, the resulting business process models with specified trustworthiness requirements can be used as basis for design and developing trustworthy software systems, applications, and even evaluation of the trustworthiness properties [6] e.g., privacy, reliability, confidentiality or integrity on an abstract level.

Business process modeling offers an appropriate abstraction level to describe trustworthiness requirements and later to evaluate trustworthiness-related risks. We describe an approach to first integrate trustworthiness requirements into a business process model. Then, we present a model-driven trustworthiness requirements refinement focusing on elements necessary for satisfying trustworthiness goals and also specifying constraints on elements of the business process (data objects, events, activities, resources etc.) to satisfy trustworthiness related qualities.

As stated in Sect. 3, there are BPMN extensions for the inclusion of different security requirements, e.g., non-repudiation, attack harm detection, integrity, and access control. There are also proposed languages for the formulation of security constraints embedded in BPMN. In all these approaches, only security requirements are incorporated into a BPMN process from the perspective of a business process analyst. In our work, we consider a broad range of trustworthiness properties rather than just security. Furthermore, there is a rationale about where these trustworthiness requirements were originating from. Our proposed approach aligns organizational (business) requirements in an adequate way with trustworthiness requirements. Our approach tackles the problem of high-level and low-level trustworthiness requirements' misalignment between the business/organizational level and the application and software service level. This should satisfy business goals as well as trustworthiness goals of the end-users. The result allows a requirements engineer to create a business process specification that represents a process along with a set of trustworthiness properties that the generated software service, or app, needs to be compliant with.

Therefore, this trustworthiness requirements specification allows the designer to make informed design decisions to put the right mechanisms into place.

4.1 Conceptual Model of the Enriching Business Process Model with Trustworthiness Requirements

We define the fundamental concepts and their relations in form of a conceptual model that is depicted in Fig. 3. The conceptual model reflects the basic concepts of our approach.

The major concept of our method for eliciting and refining trustworthiness requirements is the combination of business process modeling using BPMN and goal models (cf. Fig. 2). A trustworthiness goal is a special goal that addresses the trust concerns of users. The trustworthiness goal is satisfied by trustworthiness requirements, which can be realised by trustworthiness properties. In this paper, we focus on the part for analyzing and addressing the end-users’ trust concerns, and expressing them in terms of either BPMN elements or the extended elements for trustworthiness. For instance, interactions points, defining trustworthiness-specific activities (e.g., notifications for satisfying transparency) or defining monitoring points where we can specify which part of the process needs to be monitored at run-time and what the desired behavior is. This will serve to derive trustworthiness requirements in the form of commitments reached among the participants for the achievement of their goals.

We use the term “business process element” to distinguish between generic types of BPMN, e.g., activity, resources like human resources or data objects and concrete trustworthiness-related elements “trustworthiness element” (our extension) that can pertain to a type of BPMN elements, e.g., monitor point, interaction point and constraints.

A Threat is a situation or event that, if active at run-time, could undermine the value of trustworthiness by altering the behavior of involved resources or service in the process instance. Controls are trustworthiness requirements that aim at blocking threats. Metrics are used as functions to quantify trustworthiness. A

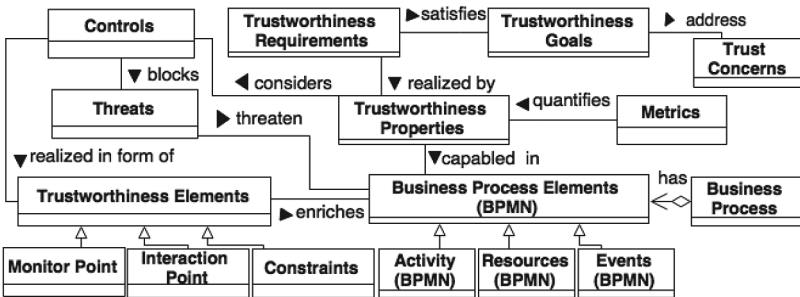


Fig. 3. The conceptual model for enriching the business process model with trustworthiness requirements

Metric is a standard way for measuring and quantifying certain trustworthiness properties and more concrete quality properties of an element [5,9].

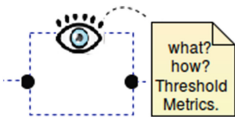
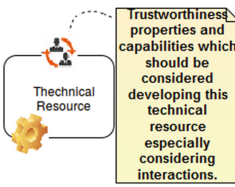
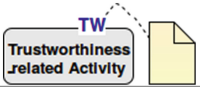
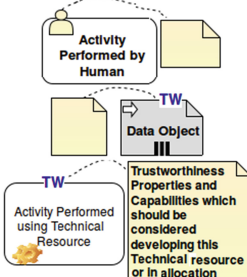
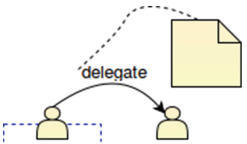
Trustworthiness elements realize the control in terms of defining elements, which directly address the trustworthiness. For instance, an additional activity can be defined to block the threat, like an activity for documenting consent or triggering a notification for a patient on delegating his/her case to another authority, or a new service from a third party is going to be used.

4.2 New Elements to Enrich the Business Process Model with Trustworthiness Requirements

We list our new elements (shown in Table 1) which are added to the business process model in BPMN to specify the trustworthiness requirements as follows:

- Monitor points: trustworthiness properties and expected behavior related to trustworthiness should be monitored. The process model must be configured before enforcing trustworthiness at run-time. We introduce the monitoring points (“*eye symbol in the model*”) with start and end points in the process model for monitoring and the trustworthiness properties that must be considered in the defined monitored points, as well as the desired/target values for them. Furthermore, the metrics can also be provided for quantifying trustworthiness properties that will be under observation at run-time.
- Interaction points: these points specify the interfaces where the end-user is involved in the business process, e.g., he/she may interact with the technical resources (e.g., apps, software services) that support him/her in performing his/her tasks. In these interfaces there are factors that could signal the trustworthiness of the system to the end-user, e.g., reliability, quality of visualization, usability, understandability of represented information, quality of service, like availability or response time. For example, if the elderly person uses an app for reviewing his/her medical plan and medication, the visualization of his/her health status and medical plan influences his/her trust about the correctness of those health reports, medications or medical plans. Therefore, the trustworthiness requirements in these points (“*interaction symbol in the model*”) need to be investigated further and the resources involved in these points should include related trustworthiness properties which satisfy the trustworthiness requirements.
- Trustworthiness constraints: in addition to new elements like monitor and interaction points, each BPMN element can be enriched/annotated with the constraints that they should keep for satisfying trustworthiness requirements. The action with trustworthiness requirements and constraints are tagged with “*TW*” in the business process model, e.g., time constraints on activities, or constraints on the resources which are used in performing a specific activity.

Table 1. Extended elements to model trustworthiness Requirements in BPMN

Defined trustworthiness element (extension)		Definition	Symbols
Monitor point		Inserting monitor points into the business process defines the start and end point of monitoring at run-time. It specifies what trustworthiness-related properties are and how they can be monitored. Monitor points can be used in combination with constraints to express the desired values and metrics for measuring trustworthiness properties at run-time	
Interaction point		Interaction points are the places where the end-user interacts with the system. The interaction is normally supported by the apps or software services. Qualities of these apps and software services have an impact on the trust perception of users. Therefore, it should be studied well how to signal their trustworthiness to the end-user. Interaction points can be further detailed in combination with constraints on those technical resources (in interaction points), e.g., specifying which quality, to what extent (e.g., 99% availability)	
Constraint	Constraints on activity	Trustworthiness requirements on a specific activity, e.g., expected duration of an activity	
	Constraints on resources	Trustworthiness requirements on a specific resource (either human or non-human), e.g., expertise of the involved human resource	
	Constraints on delegations	Trustworthiness requirements on delegation, e.g., if a delegation (e.g., activity delegation) is allowed, or delegation to whom or which roles are allowed	

5 Application Example

The example scenario presented in this work stems partially from the experience that the first author gained during the EU-funded project OPTET¹. Figure 4 shows the context of the depicted AAL scenario.

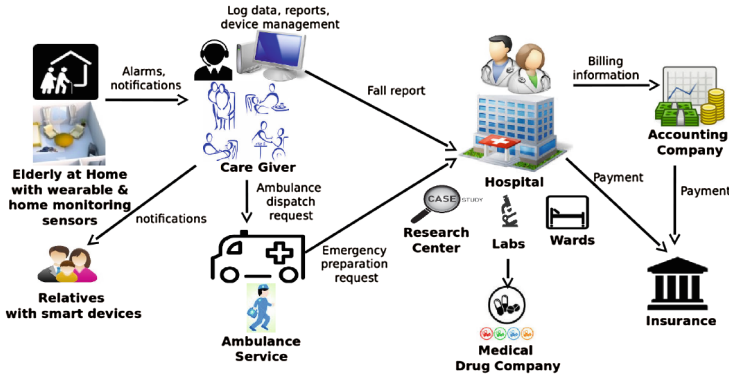


Fig. 4. The context of a home monitoring system and involved parties and actors in the scenario

The health care sector is an application area that has a lot to gain from the development of new ICT applications [1, 11]. Considering trust and trustworthiness of health care applications, one can consider a vector of multiple trustworthiness properties, which either address the fulfillment of the mission, e.g., reliability, safety, availability of the system when the patient needs help, response time of the service from the time that the patients request arrives until patient receives the needed health care, or from a privacy perspective. As an example, we consider a situation in the big picture scenario captured in Fig. 4, where the primary requirements of the patient and the requirement on the usage of elderly's data are satisfied. The elderly person, as patient, receives his/her prescribed medicine and bills are sent to the insurance company. Hence, the usage of an elderly person's data for ordering his/her medicine or payments by insurance are allowed. However, there is a secondary usage of elderly's data which violates their desired privacy level. For instance, an elderly person receives advertisements related to his/her diseases from drug companies.

Context Analysis. Here, we illustrate the high-level view of involved entities in AAL. Such AAL systems are distributed and connected via Internet in order to support the execution of the business process. The entities consist of hospital information systems, general practitioners, social centers, insurance companies, patients, their relatives, etc. Some indicative examples of electronic medical transactions are as follows:

¹ <http://www.optet.eu>.

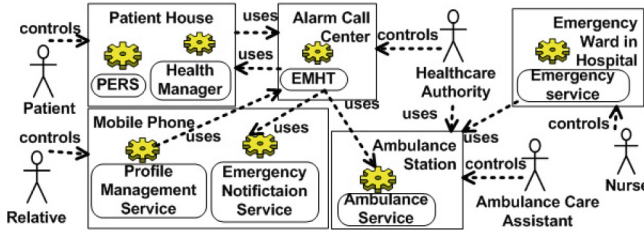


Fig. 5. Part of home monitoring system for handling healthcare cases

- Home monitoring including alarms and fall notifications,
- Emergency consultation with physician,
- Electronic notification of laboratory examination results,
- Access to the electronic medical records of patients by general practitioners,
- insurance claims.

Initially identified stakeholders in this scenario are listed below:

- End-users: here, only elderly persons are considered as end-users (cf. Fig. 1) since they are the ones that use the offered services.
- Technology providers: These are the technology providers for medical applications like software developers (cf. Fig. 1) of home monitoring systems, fall detection systems or infrastructure providers (cf. Fig. 1) like telecommunication providers, internet service providers, etc.
- Care service provider: Health care providers, health care authorities, health care centers and clinics, hospitals are physical-service providers. These are instances of organizations (cf. Fig. 1).

Our example scenario focuses on a home monitoring system for incident detection and detection of emergency cases to prevent emergency incidents from the AAL domain. Figure 5 illustrates a general approach using supporting tools and apps, to perform the activities. We assume that some of these software services are to be built by software developers, who will also benefit from the results of our work in developing a trustworthy app, software service, etc. The Fall Management System (FMS) allows elderly people in their homes to call for help in case of emergency situations. These emergency incidents are reported to an alarm call centre that, in turn, reacts by e.g., sending out ambulances or other medical caregivers, e.g., the elderly’s relatives. For preventing emergency situations, the vital signs of the elderly are diagnosed in regular intervals to reduce the hospital visits and falls.

The central asset types of the FMS include the following:

- A Personal Emergency Response System (PERS) basically consists of an alarm device which an elderly person wears so that he/she is able to call for help in an emergency situation.

- An elderly person uses the Health Manager (HM) app on his/her smart device for organizing his/her health status like requesting health service or having an overview of his/her medication, nutrition plan and appointments.
- The Alarm Call Center uses an Emergency Monitoring and Handling Tool (EMHT) to visualize, organize, and manage emergency incidents. The EMHT is a central system that receives incoming alarms from several PERS or care service requests from Health Manager apps. It gathers all relevant information related to emergency situations, health status, and supports the process of deciding and performing a certain reaction, which is performed by a human operator in an Alarm Call Center.
- An Ambulance Service is requested in case an ambulance should be sent to handle an emergency situation. The other case is that, based on analyzed information sent to EMHT, an abnormal situation is detected and further diagnoses are necessary. Therefore, the elderly person will get an appointment and notifications for a Tele-visit in his/her HM app.

Motivating Scenario. An elderly person, who lives alone in his/her apartment, does not feel comfortable after having a bad experience of a heart attack. He/she was unconscious in his/her home for several hours. The elderly person has informed the AAL services he/she considers using one of those services to avoid similar incidents in the future. Figure 6 illustrates and exemplifies the typical steps that e.g., the caregiver in the alarm center has to take once the analyzed health record of an elderly person deviates the normal situation and further examination is needed without considering trustworthiness.

The process starts by *analysing the elderly person's vital signs in the last 7 days*. These data is examined by a physician, who decides whether he/she is healthy or needs to undertake an additional examination. In the former case, the physician fills out the examination report. In the latter case, an Tele-visit is performed by this physician in which the physician informs the elderly person about examination and necessary treatment. Examination order is placed by the physician. The physician sends out a request to a clinic. This request includes information about the elderly person, and the required examination and possible labs. Furthermore, the physician arranges an appointment of the patient with the clinic for taking a sample which will be sent to the lab. Examination is prepared by a nurse of the clinic. Then, a clinic physician takes the sample. The clinic physician sends the sample to the lab indicated in the request and conducts the follow-up treatment. After receiving the sample, a lab physician validates and performs the analysis. The analysis can be done by a lab assistant. But a lab physician should validate the results. The physician from the Alarm Call Center makes the diagnosis and prescribes the medication.

Applying Proposed Approach on Motivating Scenario. Here, we demonstrate how our approach will enrich the business process model with trustworthiness requirements and then documenting those in the business process level.

Identify Trust Concerns. The elderly person is concerned about the fact whether he/she will really get the emergency help if a similar situation happens

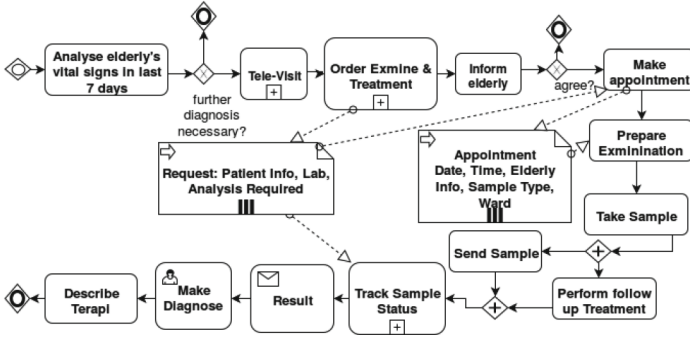


Fig. 6. Exemplary process model for analyzing elderly health situation for prohibiting emergency cases in home monitoring

again. He/she is informed that by using this service, he/she can have regular diagnoses which can reduce frequent hospital visits. However, the elderly person is concerned if he/she will be able to use the service in proper way. The elderly person is also concerned about who can get access to the data about his/her disease or life habits. He/she indicates that he/she would only like his/her regular nurse and doctor to be able to see his/her history and health status.

Identify Trustworthiness Goals. The applications of the health care domain are mission critical and privacy-related. They are mission critical, since they are monitoring the patients and dealing with the health of people. Such kinds of systems are also privacy concerned. In these systems, elderly’s data are stored, processed and communicated via Internet, where the elderly’s privacy can be threatened [23,24]. We discussed the domain and application dependence of trustworthiness properties [9]. Considering the health care domain, reliability, availability, usability, raising awareness and providing guidance to privacy and user’s data protection is a crucial issue related to trustworthiness [1, 11, 14, 16]. These are identified as trustworthiness goals addressing the identified trust concerns of the elderly person.

Our objectives are to analyse and specify trustworthiness requirements at the business process level to support the process designers and tool developers in fulfilling trustworthiness requirements and evaluating them later. Trustworthiness constraints are defined either on the resources or activities and data objects (e.g., required expertise/experience by human resource for performing an activity) or on delegation, monitor, and interaction points (cf. Table 1).

We select the business process model in Fig. 6. This business process is set up to fulfill the goal “reduce number of hospital visits”. Figure 7 illustrates the enriched business process model with the trustworthiness requirements satisfying “reliability and privacy”. Figure 7 shows the business process with the embedded trustworthiness requirements, which address the above-mentioned trust concerns. In particular, we exemplify the typical steps that a human resource (e.g., caregiver in alarm center) has to take or properties that a non-human resource

needs to have in order to contribute to trustworthiness. We start with the activity *analyse the history of the vital signs* of the elderly person in the last seven days. This activity may detect a risk in his/her health status. The following trustworthiness requirements are specified to address the trust concerns of the elderly person related to his/her confidence that he/she is not left alone and gets the needed health care in case when necessary. Furthermore, also privacy-related concerns are specified. The elderly person should receive a regular notification that informs his/her about the diagnosis and processes that are performed on his/her vital signs. This activity contributes to make him/her confident that he/she is not left alone without care. These notifications and health status reports should be comprehensible for the elderly. If a risk to his/her health status is detected, a tele-visit is offered. This activity is an interaction point supported by the HM app as technical resource (cf. Fig. 7, tele-visit activity performed by a physician). The trustworthiness properties for this interaction point are usability, response time, etc. In case of necessity for further examination he/she should be contacted by his/her physician or responsible care assistant (delegation of physician to the assistants). Furthermore, based on history, the same physician should be assigned to activities when the elderly person is in contact with the Alarm Center staff (addressing the trust concern). After processing his/her history data and if everything is alright, his/her last 7 days of vital signs should be deleted. He/she should be still informed that the process has been performed and his/her health status is fine. He/she should be informed about the deletion of his/her history as well. Figure 8 shows the refinement on the trustworthiness requirements related to “*notify elderly*” activity. The notifications and health status reports should be understandable for the elderly person. The configurability of notification mechanisms to address the usability and privacy control in terms of intervenability is addressed. Table 2 shows the trust concerns, corresponding

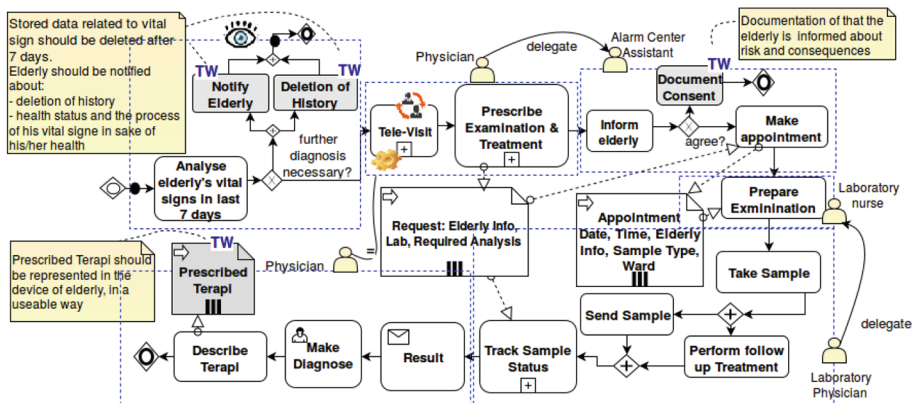


Fig. 7. Exemplary process model enriched with trustworthiness requirements and signaling controls of being worthy of trust for addressing trust concerns

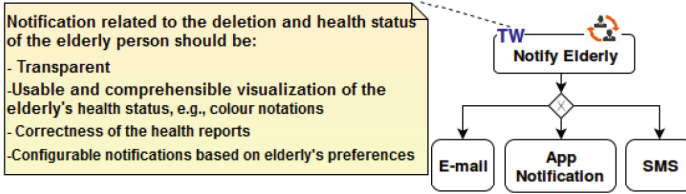


Fig. 8. Trustworthiness requirements refinement on an interaction point

Table 2. Examples of captured trustworthiness requirements and properties in the business process and directions on the design decisions

Trust concerns	Trustworthiness requirements	Activities	Affected resources
Privacy	Transparency, Intervenability	Storage, Deletion within 7 days, Update	Private inventory system from Alarm Call Center, External cloud storage
Awareness	Usability, Transparency, Reliability, Availability	Notifications, Place appointments	App on elderly's smart device (HM)
Safety, Reliability	Reliability, Availability	Raise alarm	Redundant sensors in addition to PERS
Privacy	Correctness, Usability, Availability	Make appointment, Prescribe examination	Elderly's details

requirements and activities. The column *Affected Resources* exemplifies possible software design decisions on resources.

6 Conclusions and Future Work

This paper discussed trust issues in the context of BPM. In our approach, we enable the analysis of the business process from activity, resource, and data object perspectives with respect to trustworthiness.

To the best of our knowledge, we propose a novel contribution on identifying trustworthiness requirements and integrating trustworthiness properties in business process design and preparation of verification activities that satisfies trustworthiness constraints over resource allocation and activities executions. To reduce the process designer's effort, we employ an approach for modeling trustworthiness requirements along with the business process model in BPMN. We identified the elements for specifying constraints on resources and activities

that are trustworthiness-related. Then, we specify the trustworthiness requirements and constraints for those resources and activities in the business process. A solution based on data handling conditions is used to document constraints to the usage activities. The method needs to integrate fully with a business process modeling or management application. Furthermore, the approach is supported in form of a framework to support the business process life-cycle with respect to trustworthiness. The proposed approach considers the priorities of different stakeholders. However, in this paper we do not analyze whether the different stakeholders correctly report their intentions and responsibilities in the business processes. We assume a requirement engineer has already elicited the goals of involved stakeholders based on domain knowledge. In the future, we will address these issues by a method for analysis of trustworthiness requirements using goal-oriented approaches [7]. Furthermore, the social aspects of trustworthiness will be given more attention.

This is a work-in-progress paper. The main ideas and findings will be further investigated and evaluated based on the example presented in Sect. 5. This leads to the establishment of further patterns for formulating trustworthiness requirements [8]. Our future research will focus on three important questions: (1) It is important to understand how trustworthiness properties actually influence trust. (2) We need to understand interdependencies among different trust concerns of different parties involved in the business process, and, consequently, how to define a set of trustworthiness requirements resolving conflicts. (3) Substantial work is needed to investigate existing risk assessment methodologies on the business process level, and to show how they can support business process design and building trustworthiness into the process in its whole life-cycle.

References

1. Avancha, S., Baxi, A., Kotz, D.: Privacy in mobile technology for personal health-care. *ACM Comput. Surv.* **45**(1), 3:1–3:54 (2012)
2. Becker, J., Kugeler, M., Rosemann, M. (eds.): *Process Management: A Guide for the Design of Business Processes*. Springer, Heidelberg (2003)
3. Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., Ruiz-Cortés, A.: RALph: a graphical notation for resource assignments in business processes. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) *CAiSE 2015*. LNCS, vol. 9097, pp. 53–68. Springer, Heidelberg (2015)
4. del-Río-Ortega, A., Resinas Arias de Reyna, M., Durán Toro, A., Ruiz-Cortés, A.: Defining process performance indicators by using templates and patterns. In: Barros, A., Gal, A., Kindler, E. (eds.) *BPM 2012*. LNCS, vol. 7481, pp. 223–228. Springer, Heidelberg (2012)
5. Gol Mohammadi, N., Bandyszak, T., Goldsteen, A., Kalogiros, C., Weyer, T., Moffie, M., Nasser, B.I., Surridge, M.: Combining risk-management and computational approaches for trustworthiness evaluation of socio-technical systems. In: *Proceedings of the CAiSE 2015 Forum at the 27th International Conference on Advanced Information Systems Engineering*, pp. 237–244 (2015)
6. Gol Mohammadi, N., Bandyszak, T., Kalogiros, C., Kanakakis, M., Weyer, T.: A framework for evaluating the end-to-end trustworthiness. In: *Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom)* (2015)

7. Gol Mohammadi, N., Heisel, M.: A Framework for Systematic Analysis of Trustworthiness Requirements using i* and BPMN (Submitted 2016)
8. Gol Mohammadi, N., Heisel, M.: Patterns for Identification of Trust Concerns and Specification of Trustworthiness Requirements, Accepted, in the progress of publication (2016)
9. Gol Mohammadi, N., Paulus, S., Bishr, M., Metzger, A., Könnecke, H., Hartenstein, S., Weyer, T., Pohl, K.: Trustworthiness attributes and metrics for engineering trusted internet-based software systems. In: Helfert, M., Desprez, F., Ferguson, D., Leymann, F. (eds.) CLOSER 2013. CCIS, vol. 453, pp. 19–35. Springer, Heidelberg (2014)
10. Gol Mohammadi, N., Paulus, S., Bishr, M., Metzger, A., Koennecke, H., Hartenstein, S., Pohl, K.: An analysis of software quality attributes and their contribution to trustworthiness. In: Proceedings of the 3rd International Conference on Cloud Computing and Services Science, pp. 542–552 (2013)
11. Gritzalis, S.: Enhancing privacy and data protection in electronic medical environments. *J. Med. Syst.* **28**(6), 535–547 (2004)
12. Hu, J.: Derivation of trust federation for collaborative business processes. *Inf. Syst. Front.* **13**(3), 305–319 (2011)
13. Koschmider, A., Yingbo, L., Schuster, T.: Role assignment in business process models. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 37–49. Springer, Heidelberg (2012)
14. Leino-Kilpi, H., Välimäki, M., Dassen, T., Gasull, M., Lemonidou, C., Scott, A., Arndt, M.: Privacy: a review of the literature. *Int. J. Nurs. Stud.* **38**(6), 663–671 (2001)
15. Mei, H., Huang, G., Xie, T.: Internetware: a software paradigm for internet computing. *Computer* **45**(6), 26–31 (2012)
16. Meingast, M., Roosta, T., Sastry, S.: Security and privacy issues with health care information technology. In: 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), pp. 5453–5458 (2006)
17. OMG. Business Process Model and Notation (BPMN) version 2.0. Technical report (2011)
18. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow resource patterns: identification, representation and tool support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
19. Short, S., Kaluvuri, S.P.: A data-centric approach for privacy-aware business process enablement. In: van Sinderen, M., Johnson, P. (eds.) IWEI 2011. LNBIP, vol. 76, pp. 191–203. Springer, Heidelberg (2011)
20. Stepien, B., Felty, A., Matwin, S.: A non-technical user-oriented display notation for XACML conditions. In: Babin, G., Kropf, P., Weiss, M. (eds.) E-Technologies: Innovation in an Open World. LNBIP, vol. 26, pp. 53–64. Springer, Heidelberg (2009)
21. Strembeck, M., Mendling, J.: Modeling process-related RBAC models with extended UML activity models. *Inf. Softw. Technol.* **53**(5), 456–483 (2011)
22. Sztompka, P.: Trust: A Sociological Theory. Cambridge University Press, UK (2000)
23. U.S. Department of Health and Human Services. The Health Insurance Portability and Accountability Act (HIPAA). <http://www.hhs.gov/ocr/privacy/>
24. U.S. Department of Health and Human Services. Privacy in Health Care-Standards for Privacy of Individually Identifiable Health Information (2001)

25. van der Aalst, W.M.P., Kumar, A.: A reference model for team-enabled workflow management systems. *Data Knowl. Eng.* **38**(3), 335–363 (2001)
26. Wang, M., Bandara, K., Pahl, C.: Process as a service distributed multi-tenant policy-based process runtime governance. In: *IEEE International Conference on Services Computing (SCC)*, pp. 578–585 (2010)
27. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. *J. Syst. Architect. Spec. Issue Secure SOA* **55**(4), 211–223 (2009)

A Model for Personalised Perception of Policies

Anirban Basu¹(✉), Stephen Marsh², Mohammad Shahriar Rahman¹,
and Shinsaku Kiyomoto¹

¹ KDDI R&D Laboratories, Fujimino, Japan
{basu,mohammad,kiyomoto}@kddilabs.jp

² University of Ontario Institute of Technology, Oshawa, Canada
stephen.marsh@uoit.ca

Abstract. We are often presented with policy terms that we agree with but are unable to gauge our personal perceptions (e.g., in terms of associated risks) of those terms. In some cases, although partial agreement is acceptable (e.g., allowing a mobile application to access specific resources), one is unable to quantify, even in relative terms, perceptions such as the risks to one's privacy. There has been research done in the area of privacy risk quantification, especially around data release, which present macroscopic views of the risks of re-identification of an individual. In this position paper, we propose a novel model for the personalised perception, using privacy risk perception as an example, of policy terms from an individual's viewpoint. In order to cater for inconsistencies of opinion, our model utilises the building blocks of the analytic hierarchy process and concordance correlation. The quantification of perception is idiosyncratic, hence can be seen as a measure for trust empowerment. It can also help a user compare and evaluate different policies as well as the impacts of partial agreement of terms. While we discuss the perception of risk in this paper, our model is applicable to perception of any other qualitative and emotive feature or thought associated with a policy.

Keywords: Trust · Perception · Personalised · Qualitative · Privacy · Risk · Policy

1 Introduction

As pervasive computing devices – smart watches, smart phones, tablet and personal computers – increasingly become sources of personal data, many services require users to agree with terms and conditions of usage and data sharing including access to various device features, e.g., camera, microphone and location tracking. Some of these requested features and attributes may be optional while some others may not be. Users often opt for default settings and agree with the terms and conditions without having clear understandings of what such agreements constitute.

On the other hand, organisations collecting personal data (upon agreements with users) aim to quantify privacy guarantees from macroscopic perspectives.

For instance, privacy guarantees are made about the re-identifiability of an individual from a collection of personal data that is either made public or shared with other organisations. However, one user may be more sensitive to giving away certain personal information than another user, and thus feel uneasy with generalised privacy guarantees. Macroscopic privacy guarantees are unable to capture those nuances stemming from personalised perspectives.

In this position paper, we assume that a mapping exists that can transform a policy to a set of attributes that users can understand. This may be simply a breakdown of complex legal terms into user-friendly attributes. We propose a mechanism to help users make quantitative evaluations of a policies (e.g., in terms of risks) based on criteria that the users can define. These quantitative evaluations are also expected to help users compare policies from their own perspectives. These quantifications of subjective opinion aid the trust reasoning processes at the users' ends, by enabling personalised interpretations to each user. Though quantitative, the evaluations are highly subjective and therefore the interpretation of policies cannot be compared across users.

The remainder of the paper is organised as follows. In Sect. 2, we present a brief description of related work followed by a background of the Analytic Hierarchy Process in Sect. 3. We propose our model for personalised perception (of privacy risks) in Sect. 4. We discuss the relation of this work with trust empowerment in Sect. 5 before concluding with pointers to future work in Sect. 6.

2 Related Work

When datasets containing sensitive information about individuals are released publicly or shared between organisations, the datasets go through what is known as privacy preserving data release (PPDP). Various anonymisation models, e.g., k -anonymity [1], l -diversity [2], and t -closeness [3] can be employed to minimise the possibility of re-identification of an individual from the released data. Such a re-identification poses a privacy risk. The anonymisation models used to minimise this risk typically quantifies the probability of re-identification in the theoretical worst-case scenarios. There has been work [4, 5] on modelling the risk of re-identification from empirical analysis in comparison with theoretical guarantees.

In an approach somewhat different from the aforementioned PPDP, the idea of differential privacy [6] ensures that responses to queries on data models based on sensitive data do not give away any hint from which the presence or the absence of a particular data record, pertaining to an individual, can be inferred. Privacy-preserving data mining (PPDM) aims to build various machine learning models [7–14] to ensure the privacy of the sensitive data used in building those models. Typically, the privacy is preserved through operations in encrypted domain or through perturbation of the data. The former approach has a tradeoff with efficiency due to the use of computationally intensive homomorphic encryption while the latter approach presents a tradeoff with accuracy, and thus utility of the data. None of these models cater for any personal interpretation of privacy.

In a different research strand, Murayama et al.’s work [15] surveys the ‘sense of security’ (particularly within the context of Japanese society), which is a personal perspective. Winslett et al.’s work [16,17] proposes a mechanism for trust negotiation based on interpretation of policies. Kiyomoto et al.’s work on privacy policy manager [18] discusses a framework that enables interpreting privacy policies easier for users, which has been standardised by the oneM2M initiative [19]. Kosa et al. [20] have attempted to measure privacy with a finite state machine representation. Li et al.’s work [21] attempts to make users’ privacy preferences more usable through modelling such preferences based on clustering techniques to identify user profiles.

Morton’s work [22] suggests that besides understanding privacy from a generalised level, focus should be given to individual’s privacy concern. As a first step to developing this paradigm, an exploratory study has been conducted to investigate the technology attributes and the environmental cues (e.g., friends’ advice and experiences, media stories amongst others) that individuals take into consideration. Wu et al.’s work [23] analyse users’ behaviour towards personal information disclosure with relation to the order in which personal data attributes are requested.

Stemming from the concept of privacy personalisation, in this paper, we have embarked on the quantification of subjective personal perception (of risks, or any other factors) taking into account the inconsistencies that arise when quantifying such qualitative opinion. The objective of such quantification is to give users user-centric understandings of policies and their risks, so that such understanding may assist making decisions through the concept of trust empowerment [24].

3 Background

3.1 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) due to [25] was developed in the 1970s. AHP helps with organising complex multi-criteria decision making processes. It can be used, for instance, in selecting a candidate for a vacancy based on multi-criteria evaluations in interviews. It can also be used in deciding a product to buy given various alternatives and multiple criteria for judging the alternatives. AHP can be visualised as a hierarchical structure between the goal, the selection criteria and the candidate alternatives. Figure 1 illustrates the hierarchies.

AHP assigns numeric values to the alternatives, thus facilitating a ranking. The ordering of the alternatives in such ranking is more important than the absolute numeric values associated with the alternatives. It is to be noted that in decision making problems where cost is a factor, the cost is generally not considered as a criterion in the AHP process so that each alternative ranked by the AHP can be evaluated in terms of utility-versus-cost. However, both qualitative and quantitative criteria can be used in AHP.

The relative importance of each criterion is determined at first using pairwise comparison. The integer scale [1–9] is used. For criterion X compared to Y, 1 signifies that X and Y are *equally* important, 3 signifies that X is *moderately*

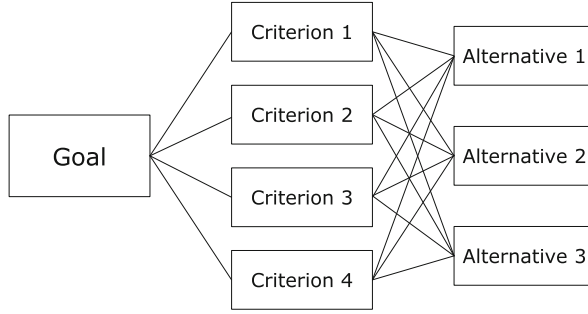


Fig. 1. The analytic hierarchy process in a diagram.

more important than Y; 5 signifies X is *strongly* more important than Y; 7 signifies that X is *very strongly* more important than Y while 9 implies that X is *extremely* important in comparison with Y. The even values 2, 4, 6 and 8 can be used to specify the intermediate values. The inverse relation is multiplicative, i.e., if $X : Y = 9$ (i.e., X is extremely important in comparison with Y) then $Y : X = \frac{1}{9}$.

This pairwise comparison is described as a matrix. Thus, for a k criteria comparison, we can have a comparison matrix \mathbf{C} of $k \times k$ elements where the leading diagonal contains elements that are all 1 (i.e., every criterion compared with itself) and the upper triangular contains elements that are the multiplicative inverses of their corresponding elements in the lower triangular, i.e., any element $C_{i,j} = \frac{1}{C_{j,i}}$ (even for $i = j$). Saty showed in [25] that the principal eigenvector of the matrix $\mathbf{C}\mathbf{c} = \lambda\mathbf{c}$ is a k -length vector \mathbf{c} , which contains the relative importances of the k elements of the criteria, which means that the criteria can now be ordered.

Having obtained a relative ordering of the criteria, each alternative is compared pairwise with each other for each criteria generating k $m \times m$ matrices given that there are m alternatives. Computing the eigenvectors of each such matrix produces a $m \times k$ matrix of relative importances of the alternatives. Multiplying this matrix with the k -length criteria ranking vector will produce a m -length vector of weighted importances of each alternatives. This helps in ranking the alternatives and thereby making a decision. To keep things simple in this explanation, we have omitted the normalisation processes and the consistency ratio, which may arise from large inconsistencies in the way pairwise comparisons are made. In our proposed scheme, we only need to make use of the relative ordering of criteria.

4 Modelling Perception

In this section, we propose a model to help users quantify, from their personal perspectives, the risks to their privacy associated with policy agreements. As mentioned earlier, we use risks as an example but the model can be applied to

any other factors too. These policies could be of different types, e.g., computer application terms of use, data release license, and so on. The key challenge in quantifying such personal perspectives is that they are highly qualitative and often inconsistent. We propose to make use of the well-known *analytic hierarchy process* to obtain quantification of subjective opinion. The quantifications are, however, indicative figures. When comparing the different policies, more importance should be attached to the relative ordering of policies than to the absolute quantitative values. The perspectives being personal, none of those quantitative figures are comparable between different users.

Running Example: To help the reader conceptualise our proposal, let us assume, as a running example without loss of generality, that the user wishes to quantify her perceptions of two applications, X and Y, on her smart phone with respect to the policy of each application defines regarding the resources it wants to access.

4.1 AHP Based Ranking of Preferences

To quantify (risk) perception of a policy, we assume that the policy has been mapped into easy-to-understand constituent parts, such that the user is able to associate some or all the parts of the policy with corresponding preferences she has in her mind. The user is required to associate free-text labels to categorise the constituent parts of the policy. Consider, for instance, a mobile phone app requesting access to the back camera, the contacts list and the microphone. One user, Alice, may have a mental model whereby she labels the access to contacts list as *contacts-access*, and views a policy asking for permission to access this as not particularly intrusive. A different user, Bob, could group the access to both the camera and the microphone with his label, e.g., *av-recording-access*, and views access to these as intrusive. Such labels are personal requiring no consistency to be preserved between labels used by different users.

Let us assume that a specific user has defined labels as a set \mathbf{L} containing k elements: $\{L_i\}_{i=1}^k$. The constituent parts of a policy may be a superset of those labels. In other words, the labels defined by the user may not be exhaustive enough to exclusively tag all the corresponding elements of a policy. This is okay because the quantification of perception would be based on what can be tagged while the rest will be ignored (although, the user will be notified of this exclusion). Having constructed some labels, the user needs an importance ranking of those labels. The importance measure can either communicate a factor (e.g., risk) directly or it may communicate the inverse. For instance, in case of the inverse of the risk, the user is least concerned if the policy asks for something that matches a particular label, thus, the importance ranking will be inverse of the risk ranking.

To develop this internal model for ranking labels, we use pairwise comparisons in the analytic hierarchy process described in Sect. 3. For simplicity yet without loss of generality, we do not take into consideration the situation where each

label may be further broken down into multiple labels, from a semantic point-of-view although we may consider this in future work. Thus, for k labels, we will need $k(k-1)/2$ or, order $\mathcal{O}(k^2)$ pairwise comparisons. If the set of labels is changed then a re-comparison is required to rebuild the label ranking. We assume that changing the label set is a relatively infrequent process. Assuming that the pairwise comparisons generate a ranking within the 10% acceptance level of the consistency ratio, the output of the AHP is a k -length preferences vector, $\mathbf{v} = \{v_i\}_{i=1}^k$, where each element L_i consists of a value v_i . These values can be used to determine the ranking of the labels.

Running Example: Let us assume that the user labels four different resources as *camera* (L_1), *microphone* (L_2), *notifications* (L_3) and *location* (L_4). We use the web-based tool at <http://goo.gl/XAulEF> to compute the ranking of our labels through AHP. The tool allows for ranking a number of items through pairwise comparisons. Each pairwise comparison is done through a sliding scale where moving the slider to one side implies preferring the item on that side of the slider to the other. The left-most point in the scale is 1 and the right-most point is 17 with 9 being neutral. In terms of AHP comparisons, the slider value of 9 signifies neutrality, i.e., neither item is preferred to the other. This corresponds to the AHP comparisons representation of $L_1 : L_2 = 1$. Moving the slider towards L_1 allows expressing the values of $L_1 : L_2$ from 2 through 9, while moving the slider towards L_2 allows representing the values of $L_1 : L_2$ from $\frac{1}{2}$ through $\frac{1}{9}$ (or inversely, of $L_2 : L_1$ from 2 through 9). The final result shows the ranked list of items, including the individual elements of the eigenvector (resulting from the AHP). The tool also helps scaling the importances of the ranks, which is beyond the scope of this paper.

Using this AHP computation tool, we express the quantification of importance in terms of risks in pairwise comparisons. Assume that the user inputs comparisons are as follows.

- $L_1 : L_2 = 6$ (14 on the slider of the AHP computation tool where 9 is in the middle signifying neutral).
- $L_1 : L_3 = 7$ (15 on the slider).
- $L_1 : L_4 = \frac{1}{3}$ (7 on the slider).
- $L_2 : L_3 = 3$ (11 on the slider).
- $L_2 : L_4 = \frac{1}{7}$ (3 on the slider).
- $L_3 : L_4 = \frac{1}{8}$ (2 on the slider).

Figure 2 shows our comparisons as done through the AHP computation tool. This sort of comparison translates to the fact that the user perceives the access to the camera 6 times as important as that to the microphone in terms of risk, while access to the location is seen as 3 times as important as that to the camera. AHP over that data generates ranking values of $L_4 = 0.5690$, $L_1 = 0.3054$, $L_2 = 0.0817$ and $L_3 = 0.0439$ with a consistency ratio of 0.086, or under 9%. Thus, we have the vector $\mathbf{v} = \{0.3054, 0.0817, 0.0439, 0.5690\}$ corresponding to location (L_1), camera (L_2), microphone (L_3) and notification (L_4), respectively. This is

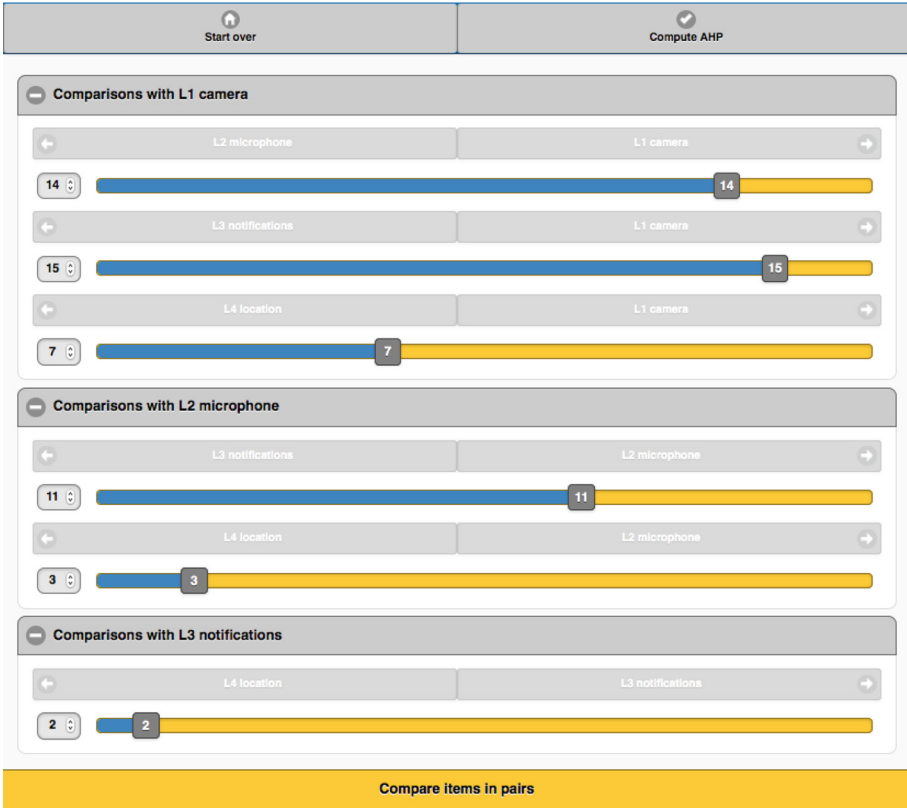


Fig. 2. The pairwise comparisons in the AHP tool showing our example comparisons.

consistent with the fact that the user views access to the location more important, with respect to privacy risk, than that to the camera and so on. Even though that conclusion may seem obvious at this point, AHP can smoothen out inconsistencies arising from the comparisons.

4.2 Optimism or Pessimism: Negative, Neutral or Positive Perception

When a new policy is encountered, some or all of its mapped terms are exclusively tagged with existing labels that the user has defined. The label-to-policy-term mapping is essentially one-to-one. Thereafter, the user is required to assign a numeric score, s_{t_i} , in a fixed positive numeric range, e.g., [1 10]. The range can be fixed once for all policies by the user. Such scores for all such policies are recorded by the local device, which can compute a centrality measure (e.g., median) for all those scores. Let us call this \bar{s} . A policy term t_i is considered to infer negative, neutral or positive perception, p_i , depending on if $t_i < \bar{s}$, $t_i = \bar{s}$

or $t_i > \bar{s}$ respectively. Perception of negative, neutral or positive bias is inspired by Marsh’s work on optimism and pessimism in trust [26]. Estimating individual term scores based on a central value \bar{s} , our model takes into account unintended biases that users have when they are asked to assign numeric scores. This process of determining perception evaluated against a k -element set of labels outputs a k -length vector of perceptions, $\mathbf{p} = \{p_i |_{i=1}^k\}$, where each label L_i corresponds to a perception p_i for a particular policy.

Running Example: Let us assume that both applications X and Y have policy terms that can be mapped exactly to the user-defined labels, i.e., *camera*, *microphone*, *notifications* and *location*. In other words, each app requires access to each of the labelled resources and each such access is specified in a policy term. Let us also assume that the user rates each such policy term using a positive numeric scale [1–10]. Suppose the user attaches the following numeric scores to terms of X: $s_{X_{t_1}} = 6$, $s_{X_{t_2}} = 7$, $s_{X_{t_3}} = 9$, $s_{X_{t_4}} = 9$ where each t_i corresponds to L_i . Thus, the median is $\bar{s}_X = 8$ and the equivalent perception vector is $\mathbf{p}_X = \{-1, -1, 1, 1\}$. Similarly, suppose the user attaches the following numeric scores to terms of Y: $s_{Y_{t_1}} = 8$, $s_{Y_{t_2}} = 7$, $s_{Y_{t_3}} = 8$, $s_{Y_{t_4}} = 9$. The median is $\bar{s}_Y = 7.5$ and the equivalent perception vector is $\mathbf{p}_Y = \{1, -1, 1, 1\}$.

4.3 Weighted Score for Policies

The perceptions of individual terms weighted by the preferences is obtained by computing a Hadamard or Schur product of the AHP-ranked preferences vector and the perceptions vector: $\mathbf{v} \circ \mathbf{p}$. An aggregate score for a policy is generated, in order to define a basis for comparison, by computing the average of the elements in the product $\mathbf{v} \circ \mathbf{p} = \{v_i p_i |_{i=1}^k\}$, i.e., a policy score $r = \frac{1}{k} \sum_{i=1}^k v_i p_i$. The closer to zero this score is, the implication is that both the positive and the negative perceptions of the policy balance out. Similarly, the more negative it is, the more negative perceptions rule; while the more positive it is; the policy contains mostly terms that the user has positive perception about.

Running Example: Based on the previously computed perception vectors, we can now define the perception score for X as follows from the Hadamard or Schur product: $r_X = ((-1) \times 0.3054 + (-1) \times 0.0817 + (1) \times 0.0439 + (1) \times 0.5690)/4 = 0.05645$. Similarly, the score for Y will be: $r_Y = (0.3054 - 0.0817 + 0.0439 + 0.5690)/4 = 0.20915$. This means that user has a more positive view of the policy terms specified by Y than those specified by X, which is in accord with the perception vectors for the policies. In both cases, positive perceptions outweigh the negative ones but X loses out to Y. The quantitative values can help the user develop an idea of how much more favourable one policy is compared to another.

4.4 (Dis)similarity Between Two Policies

A label-by-label comparison can also be done between any two policies, assuming that they correspond to the same set of labels exactly, or there exists a subset

of labels that are common to both policies. In this case, we assume that the user has a defined set of labels, $\mathbf{L} = \{L_i\}_{i=1}^k$, but the user does not need the comparison of labels themselves, i.e., no need to construct the preferences vector. Assuming that two policies, P_1 and P_2 correspond completely and exhaustively to the same set of labels \mathbf{L} , the user assigns two sets of numeric scores for each label for each policy, represented by $\{s_{1_{t_i}}\}$ and $\{s_{2_{t_i}}\}$ respectively. As before, the centrality measures of these sets are computed separately as \bar{s}_1 and \bar{s}_2 . The two policies are considered to be concordant over a term t_i if $s_{1_{t_i}} > \bar{s}_1$ and $s_{2_{t_i}} > \bar{s}_2$ or $s_{1_{t_i}} < \bar{s}_1$ and $s_{2_{t_i}} < \bar{s}_2$. They are said to be discordant over the same term if $s_{1_{t_i}} > \bar{s}_1$ and $s_{2_{t_i}} < \bar{s}_2$ or $s_{1_{t_i}} < \bar{s}_1$ and $s_{2_{t_i}} > \bar{s}_2$. They are tied for that term if $s_{1_{t_i}} = \bar{s}_1$ and $s_{2_{t_i}} = \bar{s}_2$.

A comparison of these two policies can be achieved by computing a non-parametric statistic, Somer's d, as $d = \frac{C-D}{k-T}$ where C and D are the numbers of concordant and discordant terms and T is the number of terms tied between the two policies (while k is the total number of comparable terms). The Somer's d signifies the degree of similarity of the users' perception between the two policies. Similar to the policy score, this similarity measure is not comparable across users.

Running Example: Given the perception vectors $\mathbf{p}_X = \{-1, -1, 1, 1\}$ and $\mathbf{p}_Y = \{1, -1, 1, 1\}$ both of which map to the exact same set of policy terms, we see that X and Y are concordant over terms t_2, t_3 and t_4 ; and discordant over term t_1 . To compute the Somer's d, as $d = \frac{C-D}{k-T}$, we have $C = 3, D = 1, k = 4, T = 0$. Thus, $d = \frac{3-1}{4-0} = 0.5$. A positive Somer's d indicates *similarity* between the two policies while a negative d would have implied *dissimilarity*. The Somer's d has a range of $[-1, 1]$; a value of -1 means *most dissimilar* while 1 implies *most similar*. The value of 0.5 in this case implies that the policies are somewhat similar, while the previously obtained score of each policy offers an insight into how much favourable (or not) is one policy compared to the other.

5 Trust Empowerment

We envisage that personalised perception of features, such as risk, enables users to have freedom and consistency in their thought process without having any clear idea about the policy terms. Trust is an inherently subjective phenomenon. Whilst this is often stated, it makes sense to repeat it occasionally. As we have noted before [24, 25] the systems that we and others build that 'use' trust should be seen in the light of empowerment through trust reasoning, not enforcement through mandating trust decisions. In accord with this position, we conjecture that there is a great deal to be gained from making as many parameters of the trust reasoning process as subjective and tailored to the specific user as possible. The use of subjective viewpoints of policies and the risks associated with them is a step in this direction. The hypothesis, then, is that subjective parameters increase the efficacy, tailorability and understandability of the computational

trust reasoning process and its alignment to human users. An additional hypothesis is that trust models built with such subjective notions tied to them are likely to be more robust against attacks that exploit homogeneity.

It goes without saying, of course, that intuitively there is sense here, whilst practically, much still needs to be done to confirm the intuition. Future work is planned that will work toward confirming our hypotheses, including user studies and simulations.

6 Conclusions and Future Work

In this position paper, we have introduced a novel idea for personalised quantification of emotive perceptions, such as privacy risks, associated with policies that, we believe, could assist users in making decisions through trust empowerment. The evaluation of the proposed scheme using the technology acceptance testing (TAM) and the consideration of semantic dependencies of policy terms, amongst others, are avenues of future work.

References

1. Sweeney, L.: *k*-anonymity: a model for protecting privacy. *Int. J. Uncertainty Fuzziness Knowl. Based Syst.* **10**(5), 557–570 (2002)
2. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M.: *l*-diversity: privacy beyond *k*-anonymity. *ACM Trans. Knowl. Discov. Data (TKDD)* **1**(1), 3:1–3:52 (2007)
3. Li, N., Li, T., Venkatasubramanian, S.: *t*-closeness: privacy beyond *k*-anonymity and *l*-diversity. In: *IEEE 23rd International Conference on Data Engineering, 2007. ICDE 2007*, pp. 106–115. IEEE (2007)
4. Basu, A., Monreale, A., Trasarti, R., Corena, J.C., Giannotti, F., Pedreschi, D., Kiyomoto, S., Miyake, Y., Yanagihara, T.: A risk model for privacy in trajectory data. *J. Trust Manage.* **2**(1), 1–23 (2015)
5. Basu, A., Nakamura, T., Hidano, S., Kiyomoto, S.: *k*-anonymity: risks and the reality. In: *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1, pp. 983–989. IEEE (2015)
6. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) *ICALP 2006. LNCS*, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
7. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 639–644. ACM (2002)
8. Vaidya, J., Clifton, C.: Privacy-preserving *k*-means clustering over vertically partitioned data. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 206–215. ACM (2003)
9. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. *Inf. Syst.* **29**(4), 343–364 (2004)
10. Polat, H., Du, W.: Privacy-preserving collaborative filtering on vertically partitioned data. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) *PKDD 2005. LNCS (LNAI)*, vol. 3721, pp. 651–658. Springer, Heidelberg (2005)

11. Yu, H., Jiang, X., Vaidya, J.: Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 603–610. ACM (2006)
12. Laur, S., Lipmaa, H., Mielikäinen, T.: Cryptographically private support vector machines. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 618–624. ACM (2006)
13. Amirbekyan, A., Estivill-Castro, V.: Privacy preserving *DBSCAN* for vertically partitioned data. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 141–153. Springer, Heidelberg (2006)
14. Basu, A., Vaidya, J., Kikuchi, H., Dimitrakos, T., Nair, S.K.: Privacy preserving collaborative filtering for SaaS enabling PaaS clouds. *J. Cloud Comput.* **1**(1), 1–14 (2012)
15. Murayama, Y., Hikage, N., Hauser, C., Chakraborty, B., Segawa, N.: An anshin model for the evaluation of the sense of security. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. HICSS 2006, vol. 8, pp. 205a–205a. IEEE (2006)
16. Winslett, M., Yu, T., Seamons, K.E., Hess, A., Jacobson, J., Jarvis, R., Smith, B., Yu, L.: Negotiating trust in the web. *IEEE Internet Comput.* **6**(6), 30–37 (2002)
17. Lee, A.J., Winslett, M., Perano, K.J.: TrustBuilder2: a reconfigurable framework for trust negotiation. In: Ferrari, E., Li, N., Bertino, E., Karabulut, Y. (eds.) IFIPTM 2009. IFIP AICT, vol. 300, pp. 176–195. Springer, Heidelberg (2009)
18. Kiyomoto, S., Nakamura, T., Takasaki, H., Watanabe, R., Miyake, Y.: PPM: privacy policy manager for personalized services. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES Workshops 2013. LNCS, vol. 8128, pp. 377–392. Springer, Heidelberg (2013)
19. Datta, S.K., Gyrard, A., Bonnet, C., Boudaoud, K.: oneM2M architecture based user centric IoT application development. In: 2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud), pp. 100–107. IEEE (2015)
20. Kosa, T.A., ei-Khatib, K., Marsh, S.: Measuring privacy. *J. Internet Serv. Inf. Secur. (JISIS)* **1**(4), 60–73 (2011)
21. Lin, J., Liu, B., Sadeh, N., Hong, J.I.: Modeling users mobile app privacy preferences: restoring usability in a sea of permission settings. In: Symposium On Usable Privacy and Security (SOUPS 2014), pp. 199–212 (2014)
22. Morton, A.: “All my mates have got it, so it must be okay”: constructing a richer understanding of privacy concerns an exploratory focus group study. In: Gutwirth, S., Leenes, R., De Hert, P. (eds.) Reloading Data Protection, pp. 259–298. Springer, Heidelberg (2014)
23. Wu, H., Knijnenburg, B.P., Kobsa, A.: Improving the prediction of users disclosure behavior by making them disclose more predictably? In: Symposium on Usable Privacy and Security (SOUPS) (2014)
24. Dwyer, N., Basu, A., Marsh, S.: Reflections on measuring the trust empowerment potential of a digital environment. In: Fernández-Gago, C., Martinelli, F., Pearson, S., Agudo, I. (eds.) Trust Management VII. IFIP AICT, vol. 401, pp. 127–135. Springer, Heidelberg (2013)
25. Saaty, T.L.: *The Analytic Hierarchy Process: Planning, Priority Setting, Resources Allocation*. McGraw-Hill, New York (1980)
26. Marsh, S.: Optimism and pessimism in trust. In: Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA94) (1994)

Evaluation of Privacy-ABC Technologies - a Study on the Computational Efficiency

Fatbardh Veseli^(✉) and Jetzabel Serna

Chair for Mobile Business and Multilateral Security, Goethe University Frankfurt,
Theodor-W.-Adorno-Platz 4, 60323 Frankfurt, Germany
{fatbardh.veseli, jetzabel.serna-olvera}@m-chair.de

Abstract. Privacy-enhancing attribute-based credential (Privacy-ABC) technologies use different cryptographic methods to enhance the privacy of the users. This results in important practical differences between these technologies, especially with regard to efficiency, which have not been studied in depth, but is necessary for assessing their suitability for different user devices and for highly dynamic scenarios. In this paper, we compare the computational efficiency of two prominent Privacy-ABC technologies, IBM's Idemix and Microsoft's U-Prove, covering all known Privacy-ABC features. The results show that overall presentation is in general is more efficient with Idemix, whereas U-Prove is more efficient for the User side (proving) operations during the presentation, and overall when there are more attributes in a credential. For both technologies we confirmed that inspectability, non-revocation proofs, and inequality predicates are costly operations. Interestingly, the study showed that equality predicates, the number of attributes in a credential, and attribute disclosure are done very efficiently. Finally, we identified a number of specific trust issues regarding Privacy-ABC technologies.

1 Introduction

Nowadays, electronic service providers continuously collect, integrate and analyse huge amounts of personal data. This data is commonly used to provide authentication, personalized services, targeted advertisements, and to develop innovative applications that address critical societal challenges (e.g., transportation and eHealth). Nevertheless, each piece of information is a digital footprint of our identity, threatening the privacy of customers. In this regard, regulation, such as the European Data Protection Regulation [1] and the eIDAS Regulation [2] are useful instruments for protecting the privacy of individuals, but need to be further enforced with technical privacy protection mechanisms.

Privacy-enhancing Attribute-Based Credentials are innovative technologies that provide privacy-respecting authentication and access control for the customers of trust-sensitive digital services in general. Furthermore, they may relieve the Service Providers from the liability with respect to the personal data about the users by minimizing the amount of the personal data collected. The main concepts behind Privacy-ABC technologies have initially been introduced by David

Chaum’s anonymous credential systems in the 80s [3]. They enable pseudonymous access to services, minimal disclosure of attributes, and unlinkability of users’ transactions. However, despite their potential and their apparent technical maturity, their adoption is still low [4].

In this paper, we focus on the practical efficiency of the operations of Privacy-ABC technologies, which we consider to influence their adoption and suitability for deployment in a wide range of digital services. Especially when deployed for devices with limited resources (e.g., smart phones or smart cards) and with high mobility (e.g., vehicular on-board units) requirements, it is important to understand which technology performs better and which privacy features are more time-consuming for the users. Therefore, we practically compare the computational efficiency of two prominent implementations of Privacy-ABC technologies, namely Microsoft’s U-Prove [5] and IBM’s Identity Mixer (Idemix) [6]. We further evaluate the cost advanced features on the efficiency, such as inspectability, non-revocation proofs, predicates, number of attributes, and number of disclosed attributes. Finally, we analyze the implications of these technologies in digital services and identify important trust considerations that are also important for the further adoption of the technology.

The rest of the paper is organized as follows, Sect. 2 introduces the main concepts of Privacy-ABC technologies. Section 3 presents the related work. Section 4 introduces the study methodology, whereas the main results are presented in Sect. 5. These results are then discussed in Sect. 6, where also important challenges and important trust considerations are identified. Finally, Sect. 7 concludes the paper.

2 Background

A general architecture of Privacy-ABC technologies consists of three main entities, namely a *User*, an *Issuer*, and a *Verifier*. The Issuer issues *credential(s)* to the *User*, which can later be used for authentication with the Verifier. A credential contains attributes about the User, e.g. name, date of birth, etc. Such an architecture may optionally also include an entity that takes care of revocation of credentials, and another entity that can revoke the anonymity of otherwise anonymous users. The main interactions between Privacy-ABCs system entities [7] can be represented by the different stages of its lifecycle, that is, *issuance*, *presentation*, *inspection*, and *revocation*.

- **Issuance** of a credential is an interactive protocol between a User and an Issuer. By issuing a credential to the User, the Issuer vouches for the correctness of the attribute values contained in the credential.
- **Presentation** is an interactive protocol in which the User proves the possession of certain credential(s) or claims about the attributes. A *presentation token* is a cryptographic proof derived from the (credential of the) User as an evidence of possessing certain credential(s), optionally disclosing some attribute to the Verifier.

- **Inspection** provides conditional anonymity. It enables a trusted entity (i.e., an Inspector) to revoke the otherwise anonymous transaction (presentation).
- **Revocation** ends the validity of the credentials whenever necessary, such in case of service misuse, credential compromise, or loss of credential storage medium (e.g. smart card).

At the core of Privacy-ABCs untraceability and unlinkability of credentials are the two most important privacy-related properties. *Untraceability*¹ property which guarantees that the presentation of credential(s) cannot be linked to their issuance, whereas *unlinkability*² property guarantees that a Verifier cannot link different presentations of a given user. Additionally, Privacy-ABCs also support the following features³:

- **Carry-over of attributes** enables users to carry over some attribute(s) from an existing credential into a new one without disclosing it to the Issuer.
- **Key binding** binds one or more credentials to the same secret (protecting against credential pooling);
- **Selective disclosure** of attributes during the presentation;
- **Predicates over attributes** enables logical operators, such as “greater” or “smaller than” to be applied on hidden attributes;
- **Pseudonyms** enable pseudonymous access to services;
- **Inspectability** is an accountability feature that enables revocation of a user’s anonymity if certain pre-defined conditions are met.

3 Related Work

Efficiency of Privacy-ABC systems has been identified as an important challenge and previously discussed in a number of studies [8–12]. On a theoretical level, Baldimtsi and Lysyanskaya [12] as well as Camenisch and Groß [10] have specially addressed the importance of computational efficiency in resource-constrained devices, such as smart phones and smart cards.

Later, Camenisch and Lysyanskaya [11] proposed a signature scheme with more efficient protocols based on the Strong RSA assumption. Following this direction, Chase and Zaverucha [13] have proposed an approach for providing (a subset) features of Privacy-ABCs based on the use of message authentication codes (MACs) instead of public keys for better efficiency. However, their proposal has an important limitation since the Issuer and Verifier share the same secret key, making them not be suitable in scenarios, such as ad-hoc networks, where a regional road authority will act as an Issuer, and a number of road side units for traffic management will act as Verifiers.

A number of other efforts to achieve efficient implementations of Privacy-ABC technologies have emerged, especially focused on smart cards [8, 9, 11, 14].

¹ Also known in the literature as “Issuer-unlinkability”.

² Also known in the literature as “Verifier-unlinkability”.

³ However, Privacy-ABC technologies pose particular implications on trust assumptions in digital services, which are mentioned in Sect. 6.3.

For instance Bichsel et al. [8] reported the first practical implementation based on Idemix on a JCOP card. Mostowski and Vullers [15] optimized the efficiency U-Prove, and later Vullers and Alpar [16] optimized the efficiency for Idemix on a MULTOS card and presented a comparison of both approaches. However, their practical evaluation of Privacy-ABC technologies covered only the basic presentation of a single credential.

De la Piedra *et al.* [17] provided additional optimizations for Idemix on smart cards by implementing an efficient extended Pseudo-Random Number Generator (PRNG). Contrary to Vullers and Alpar [16], De la Piedra *et. al* presented efficiency results considering a more advanced setup which included a combination of credentials and the use of predicates. However, the authors did not cover advanced features such as inspection or revocation, and furthermore, their experiments only considered Idemix.

In summary, existing efforts have so far either focused on a single technology or covered only a limited subset of the Privacy-ABC features. We fill this gap by evaluating the computational efficiency of the two Privacy-ABC technologies covering all of the known features of Privacy-ABCs. This complements previous published work focused on storage and communication efficiency [18]. To our knowledge this is the first attempt to compare both technologies, under a common architecture and evaluation framework. This is especially important considering that the publicly available description of U-Prove or Idemix define efficiency only in theoretical terms and do not provide practical benchmarks of the actual efficiency [6, 19].

4 Methodology

In this work, we have adopted the Privacy-ABC technologies evaluation framework proposed by Veseli *et al.* [20]. This framework defines a rich set of criteria for benchmarking Privacy-ABC technologies based on their efficiency, functionality, and security assurance. With regard to efficiency, the framework distinguishes between *computational efficiency*, measured in time units; *communication efficiency*, measuring the sizes of the dynamically generated data; and *storage efficiency*, measuring the sizes of the static data in permanent storage. We focus on the former, covering all Privacy-ABC features.

Evaluated Technologies The core building block of Privacy-ABC technologies is the signature scheme. Therefore, we compare Microsoft’s U-Prove based on Brands’ signatures [21], and IBM’s Idemix, which is based on Camenisch-Lysyanskaya’s signatures [11]. As such, both technologies support selective disclosure of attributes, pseudonyms, and untraceability.⁴ Other advanced features, such as non-revocation proofs, inspectability, or predicates are supported by additional building blocks, which are shared for both Idemix and U-Prove. For

⁴ Unlike Idemix, U-Prove tokens are untraceable, but linkable between different presentations.

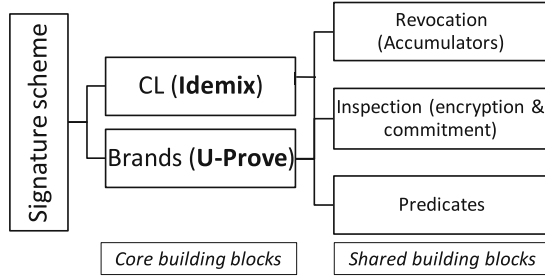


Fig. 1. Overview of the evaluated technologies

Table 1. Testbed for the experiment

Processor	1.8 GHz Intel Core i7
Number of processors	1
Number of Cores	2
L2 Cache (per Core)	256 KB
L3 Cache	4 MB
Running Memory	4 GB 1333 MHz DDR3
OS	Mac OS X

revocation, we have used the accumulator technology based on the Camenisch-Lysyanskaya [22], whereas inspectability is implemented using the verifiable encryption scheme introduced by Camenisch-Shoup [23]. Figure 1 shows an overview of the components evaluated in our study.

Contrary to the Privacy-ABC technology introduced by Persiano [24], both U-Prove and Idemix provide a similar level of technology readiness [25] and have been integrated under a common architecture [26], which has a reference implementation openly available on Github [27]. We performed the practical benchmarks using this implementation, enabling the same measurement instrument to test both technologies, providing a fair comparison.

Experimental Setup The experimental setup has been done using Java, the experiments have been executed on a computer with the configuration shown in Table 1. All experiments have been evaluated using a key length of 1024 bits, based on the RSA group⁵, which is an important element in the evaluation (see more on this in Sect. 6). All the experiments reflect the average performance time from 50 runs of each operation.

⁵ It is worth to note that the U-Prove’s implementation was instantiated over standard subgroup, alternatively it could also be based on elliptic curves. However, this was not available for the reference architecture we used.

Limitations Our results are based on the openly available versions of U-Prove and Idemix. However, U-Prove could be instantiated over elliptic curves, which can be more efficient. Furthermore, our architecture allows flexible changes in the policies, but it also represents an overhead, which could be avoided in scenarios where flexibility is not needed.

5 Results

This section provides a comparison of the computational efficiency between U-Prove and Idemix, and an evaluation of the computational cost of advanced Privacy-ABC features.

5.1 Comparison of the Efficiency of Privacy-ABC Technologies

The comparison between Idemix and U-Prove will follow the lifecycle of the credentials, covering both issuance and presentation of Privacy-ABCs. Issuance efficiency is important for cases that require periodic issuance of new credentials, whereas presentation efficiency is assumed to be important for most of the practical scenarios. A non-efficient presentation may (negatively) influence users' experience and consequently their perception on the technology, which we assume to play a crucial role on their acceptance by users [28].

Comparing Issuance Efficiency. Privacy-ABC technologies can support *simple* and *advanced* forms of issuance. In the case of simple issuance, the Issuer does not require the User to present any existing credential or pseudonym. This can be, for instance, when this is the first Privacy-ABC credential that the User gets. Advanced issuance follows a presentation of User's existing credential or pseudonym, binds a credential to the secret key of an existing credential or pseudonym (*key binding*), or even *carry over attributes* from the existing credential into the new one, without the Issuer learning their value(s).

Figure 2 compares the computational efficiency of Idemix and U-Prove for different types of issuance, where all issued credentials contain 5 attributes. It includes efficiency results for the following issuance types:

Simple issuance is the simplest and therefore the most efficient form of issuance.

It does not require the user to present any proof with Privacy-ABC technologies. For this type of issuance, both Idemix and U-Prove have a similar efficiency, where the credential is issued in less than 300 ms, with Idemix being only slightly more efficient (about 20 ms).

Show Pseudonym shows the efficiency for an advanced issuance that requires the User to present an existing pseudonym, after which the issuance of the new credential follows. The cost of showing a pseudonym is reflected in about 70% for Idemix and 80% overhead for U-Prove relative to simple issuance. Compared with U-Prove, Idemix is more efficient for about 50 ms.

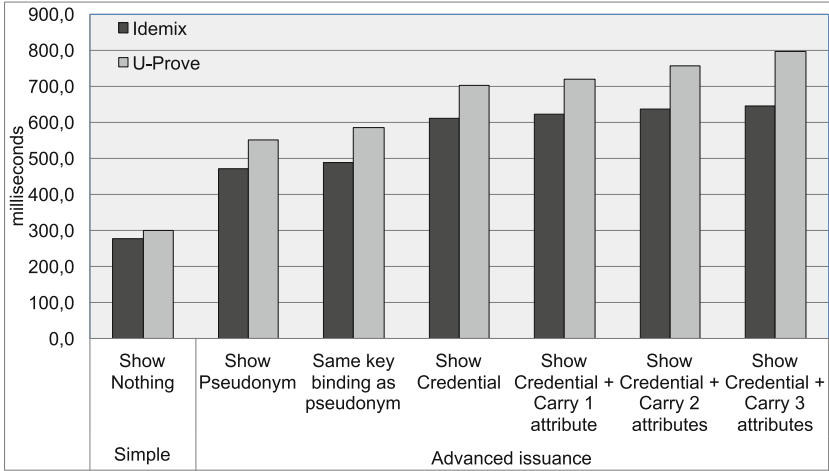


Fig. 2. Comparing the computational efficiency of Idemix and U-Prove for simple and advanced forms of issuance

Same key as pseudonym requires the new credentials to be bound to the same secret key as the pseudonym that the User presents. The effect of “key binding” has a small to small overhead for both Idemix (about 10 ms) and U-Prove (15 ms).

Show credential shows the time to get a credential issued when an existing credential is required (presented). The results show that, compared to simple issuance, this issuance has about 315 ms or 110% overhead for Idemix, and about 400 ms or 130% overhead for U-Prove. Idemix is in general more efficient than U-Prove for less than 100 ms.

Show credential + carry 1/2/3 attributes show the overhead of “carrying” 1, 2, and 3 attributes respectively for Idemix and U-Prove. For Idemix, each carried attribute represents a computational overhead of less than 15 ms, whereas for U-Prove this costs about 25 ms. Also here, Idemix is more efficient than U-Prove for 100–150 ms.

Comparing Presentation Efficiency. In Fig. 3, we provide a comparison of the computational efficiency of presentation for Idemix and U-Prove for four different presentations. We have distinguished between two steps during the presentation phase: *proving* includes the cryptographic operations performed by the User in order to generate the proof (presentation token), and *verification*, which is the step performed at the Verifier side upon receiving the presentation token of the User. This may be important, for instance, in order to distinguish between the effort distribution between the User and the Verifier for the two technologies, and adapt their computational power accordingly in order to achieve a better efficiency. The figure shows the following four presentation cases:

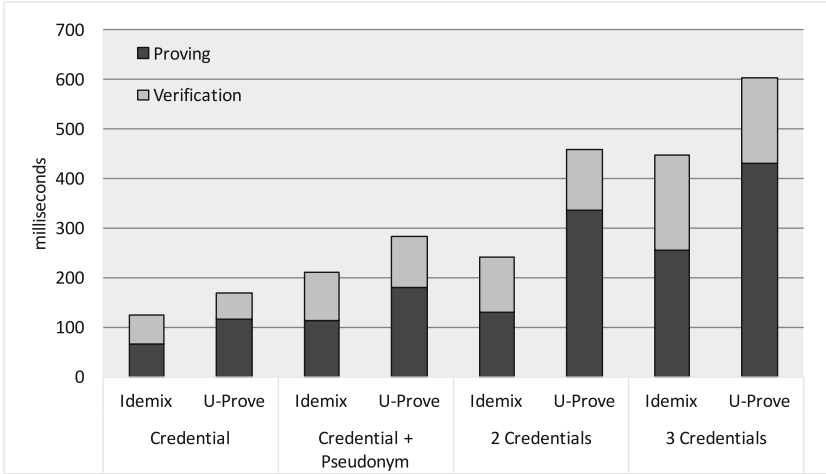


Fig. 3. Comparison of the computational efficiency of presentation of key-bound credentials between Idemix and U-Prove

Credential shows the efficiency of presenting a credential (zero knowledge) when no attributes are disclosed. The credential is key-bound, meaning that it underlies a secret key, and contains five different attributes. This type of presentation takes about 120 ms for Idemix and 180 ms for U-Prove, making Idemix slightly more efficient.

Credential + Pseudonym shows the efficiency of presenting a credential and a pseudonym, both bound to the same secret key. This can be useful, for instance, when the Verifier wants to offer the possibility to the User to maintain a “reputation” under a certain pseudonym besides having a proof of a credential. Compared to “Credential”, we can observe an overhead of showing a pseudonym, which is about 80 ms (about 60 %) for Idemix, whereas for U-Prove it represents an overhead of about 110 ms (65 %). Compared to showing a new credential, the overhead of showing a pseudonym is smaller for both technologies for 30 ms for Idemix and about 190 ms for U-Prove, which shows that presenting a pseudonym is more efficient with Idemix.

2/3 Credentials shows the overhead of additional credentials. Compared to “Credential”, the presentation time grows linearly for each presented credential for both Idemix and U-Prove. Considering that Idemix is more efficient than U-Prove for about 75 ms per credential or about 35 %, the same difference grows linearly with each new credential.

Effort Distribution. Finally, we can observe that for both technologies, proving is more costly than verification. On average, proving takes about 55 % of the total presentation time for Idemix, while for U-Prove it takes about 70 %. This is another advantage for Idemix, since the goal is to make the computation at the User part more efficient.

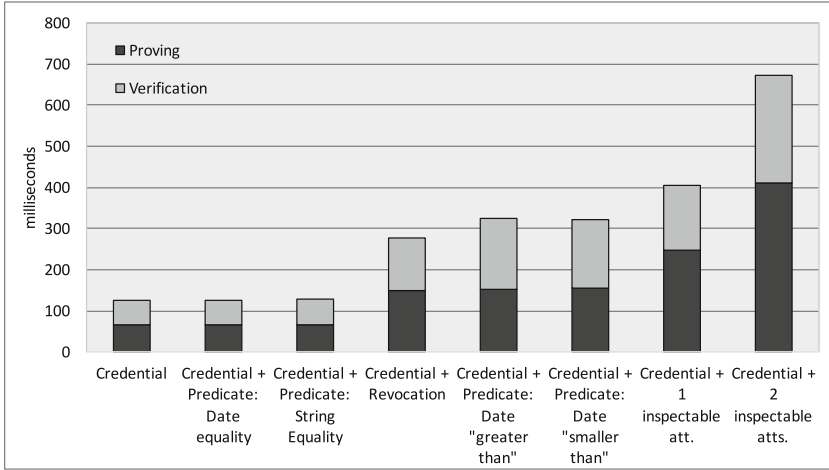


Fig. 4. Impact of the different features on the presentation efficiency (Idemix)

Additional Results. The *number of attributes* in a credential and *attribute disclosure* result in a small overhead on the presentation time for both technologies. For the former, U-Prove gets more efficient than Idemix as the number of attributes increases. Proving a credential with 5 more attributes showed better efficiency for U-Prove than for Idemix, where each new attribute cost about 4, respectively 6,5 additional ms. With regard to attribute disclosure, we noticed a small, but positive impact for both technologies, where the presentation time was about 2–5 ms more efficient for each disclosed attribute.

5.2 Evaluation of the Cost of Advanced Privacy-ABC Features

Besides the core features of Privacy-ABC technologies, such as unlinkability, selective disclosure, and pseudonyms, additional privacy features can be added to both technologies. The implications of these features on the trust issues are explained in Sect. 6, whereas this section presents the impacts of these features on the performance efficiency. The extra features correspond to specific building blocks that need to be integrated with the respective Privacy-ABC technologies, including *predicates*, *showing non-revocation*, *inspectability*, etc. The respective impact (overhead) of these features on the computational efficiency of presentation is shown in Fig. 4 for Idemix.⁶

The figure shows the computational efficiency for the following presentations (all of which require a credential of 5 attributes):

⁶ The features used in this section utilize the same “extensions”, making the computational overhead the same for both technologies. For simplicity, we show the impact on Idemix.

Credential is the basic benchmark, which simply requires presenting a credential, and serves as a reference for assessing the additional overhead of using other features.

Credential + Predicate: Date equality represents the presentation of a credential and checking that one of the attribute values (in this case, the date of birth) equals a given constant value, both of which are of type *Date*. We can observe from the results of this diagram that date equality proof represents little to no overhead on the efficiency of presentation.

Credential + Predicate: String equality is similar to the previous one, except that the compared attributes are of different data type, namely they are of type *String* (as compared to the previous one, which is *Date*). Also in this case we observe little to no overhead on the presentation time.

Credential + Revocation shows the overhead of presenting a revocable credential. In this case, the presentation requires also proving that the credential is not revoked. We can observe that the overhead of proving non-revocation accounts for about additional 160 ms or about 130 % more time (compared to “Credential”).

Credential + Predicate: Date “greater than” shows the efficiency of both presenting a credential and showing that one of the attributes (the date of birth) is greater than a certain constant value. This is especially useful for scenarios where checking the age of a person is necessary, e.g. checking that a person is not older than a given age. We can observe that, compared to “Credential”, showing that the date is greater than a given constant date costs about 200 additional ms or about 165 % more (compared to “Credential”).

Credential + Predicate: Date “smaller than” similar to the previous one, the efficiency of this predicate is comparable to checking “greater than”, i.e., about 200 additional ms or 165 % overhead. This is especially important for checking that a person is older than a given age, e.g. that a person is over 18 years old.

Credential + 1/2 Inspectable atts. shows the overhead of having one, respectively two attributes inspectable during the presentation. Typically, having one attribute inspectable should be enough in order to uncover the identity of the person in a given transaction (revoke anonymity), however there may be cases when more attributes are to be inspectable. From the figure, we can see that inspectability has the biggest overhead on the presentation among all of the presented features of Privacy-ABCs, and grows linearly by about 275 ms or by 210 % for each inspectable attribute.

6 Discussion

This section discusses important implications of the results, identifies open research challenges and important trust issues for Privacy-ABC technologies.

6.1 Implications

Results show that both technologies (U-Prove and Idemix) present similar computational efficiency for simple issuance (i.e., less than 300 ms), Idemix

outperforming U-Prove by 20 ms. However, for advanced issuance (e.g., carry over of attributes) differences could be of 150 ms, again Idemix being more efficient. Although, in many scenarios issuance efficiency will not play a major role, it may be relevant to those scenarios where users frequently need to get credentials issued interactively. For instance, in vehicular networks, the nodes are expected to send messages every 300 ms and have a communication range of 1000 m, making a time difference of 150 ms (e.g. the overhead of advanced issuance) an important decision factor for deciding the type of issuance. Nevertheless, in cases where issuance of credentials is assumed to be done off-line, the efficiency of both technologies can be considered acceptable in the aforementioned scenario.

In the following, we discuss some of the main implications related to the different features of the presentation.

Inspectability is the most computationally expensive feature that linearly grows with each new attribute made inspectable. The reason for this is the use of cryptographic *commitments* of the given attribute and *verifiable encryption* of that commitment with the public key of the Inspector. The Verifier is then able to check that the encrypted value corresponds to the value indicated and that is encrypted with the right key. However, one can assume that this process requires additionally administrative controls and therefore its efficiency is not critical in most scenarios.

Revocability is an important feature, but it is as costly as the presentation of a credential. Our revocation technology is based on the *accumulator* scheme of Camenisch-Lysyanskaya [22]. The overhead for the User comes from the fact that each revocable credential needs to be proven that it is part of the accumulator, making it not suitable for highly dynamic scenarios. A recommendation would be that, whenever more credentials are needed in a presentation, only one should be checked for revocation, and have the other credentials as non-revocable. In this way, the cost of proving non-revocation for the other credentials is avoided.

Predicates are costly when non-equality has to be shown, e.g. showing that an attribute value is greater or smaller than another one without revealing the attribute value. However, equality predicates seem to be very efficient and add very little overhead on the presentation, but they should be carefully used so that privacy is still preserved.

The number of attributes in a credential, as well as the **number of disclosed attributes** have shown small overhead on the efficiency of presentation for both, but slightly bigger overhead for Idemix, making U-Prove favourable when a credential has more attributes. Besides this, another implication for both technologies would be that a more efficient system design could result from decreasing the number of credentials whilst increasing the number of attributes in a credential whenever suitable, and disclosing only a desired subset of those attributes.

6.2 Open Challenges

Despite their practical differences, both Privacy-ABC technologies face some open challenges, calling for additional research efforts related to the efficiency,

especially with regard to their deployability in different platforms, effective revocation, and the impact of the security level on the efficiency.

Deployment Platforms. The current results are performed on a personal computer with average computational power. Deploying them on other platforms, such as smart cards, mobile devices, or in the cloud comes with specific challenges. *Smart cards* require native support the cryptographic operations, and optimisations to make them practically efficient. For *mobile devices*, there are more possibilities, but having a cross-platform solution is challenging. One way is to use Javascript or have native browser support for digital signature schemes, which are still considered challenging [29]. A *cloud* solution would ease the availability in different user devices, but creates new privacy risks, since the cloud service provider would need to be trusted. Alternatively, one should check the feasibility of integrating technologies, such as proxy re-encryption schemes [30], where the cloud service provider can act on behalf of the User without seeing the attributes in clear.

Revocation and Non-revocation Proof. There is currently a compromise between efficiency and effective revocation. Unlike in the X.509 case, with Privacy-ABCs the User should not disclose a credential identifier to check for revocation. Instead, the user must show non-revocation in zero knowledge, but this is still a costly operation for the User. This requires the User to be on-line, which is an additional limitation (makes the technology not usable on “off-line” scenarios, e.g. smart cards). One way to optimise this process could be by shifting part of the proving effort from the User to the Verifier, or to extend the periods between different revocation checks, e.g. daily or weekly, depending on the scenario.

Key Length and Efficiency. The results in the paper reflect the 1024 bit key size for both, which seems to provide comparable level of security. According to the ECRYPT report on key sizes [31] this corresponds to the symmetric key size of around 72 bits, providing “short-term protection against medium organizations, medium-term protection against small organizations”. According to the same report, an RSA cryptographic key size of 2048 bits would provide a security level corresponding to a symmetric key of around 105 bits, which is between a “legacy standard level” and one that offers a “medium-term protection” (about 10 years). Based on our experiments, the computational efficiency drops on average by a factor of four with the doubling of the key size. Therefore, an additional challenge remains to provide higher level of security assurance with smaller impact on the efficiency.

6.3 Trust Considerations for Privacy-ABC Technologies

The upcoming General Data Protection Regulation [1] requires data protection by design and default, where Privacy-ABC technologies could very well fit. One of the benefits of Privacy-ABC technologies is the fact that the User need to reveal minimal amount of information to Service Providers. Trusting on the technology

means that the users need to put less trust on the trustworthy behaviour of service providers, increasing the overall level of trust on the digital services. As already acknowledged in previous research, trust is an important element on the privacy concerns and on the level of disclosure of personal information on electronic commerce website [32] and that more trust leads to less privacy concern [33]. However, one has to ensure that technology is indeed used in a trustworthy manner. In this regard, three important considerations need to be made, which are briefly discussed in the following section.

Trusting the Verifier. Privacy-ABC technologies *enable* minimal disclosure of attributes, so that Service Providers (Verifiers) only require disclosure of the attributes that are necessary for authentication in a given scenario. This is defined in a so-called presentation policy by the Verifier based for a particular service. A malicious Verifier can define an “unfair” policy, asking the User to disclose excessive amount of attributes [34]. In such a case, there is little benefit for the privacy of the customers by using Privacy-ABC technologies. Therefore, to increase trust, there must be a way to protect from such a misuse potential, such as requiring certification of presentation policies by an external trusted entity, or providing standard presentation policies for typical use-cases [34].

Trusting the Inspector. The role of the Inspector is to revoke of anonymity of misbehaving users. While this is done to provide conditional accountability for all the cases that could “go wrong”, e.g. when a user violates the code of conduct for a given service, or when there is a threat to the lives or properties by anonymous users, this feature is somewhat controversial, as it requires trust on the proper conduct by the Inspector. Therefore, in practice, we should limit the potential for authority misuse by a malicious Inspector, who in the worst case, could revoke the anonymity of (identifying) any user without a proper ground to do so. Such a limitation could in practice be achieved by a combination of organisational processes or technical solutions. An organisational solution could be to define an organisational process that requires the approval by a committee or a group of people within an organisation for inspection. A technical solution would be to split the “inspection” key into several parts, and requiring at least two of the members of such a committee to come together to join their key parts in order to be able to perform inspection.

Trust Implications in the Cloud. Efficiency and mobility of the customers could be improved if the credentials are stored and computed on the cloud. However, this implies additional trust is needed on the cloud service provider in proper handling of user credentials. A malicious cloud service provider could then spy on the user by tracking activities, or even worse, impersonate the User without his or her consent. While there are technical potentials for limiting such cases, including the use of special cryptographic tools such as proxy re-encryption schemes [30] or similar, there is a compromise between the level of risk the users gets exposed to and the convenience of the mobility provided by the cloud.

7 Conclusion

This paper presented an evaluation of two major Privacy-ABC technologies with regard to efficiency following a common evaluation framework. Our results have shown that U-Prove is more efficient than Idemix for the User operation (proving) and in general when a credential has more attributes. However, Idemix is more efficient in the rest of the cases, especially when advanced presentation features are used. Regardless of the efficiency, Idemix also provides unlinkability between many presentations, making it more favourable in scenarios with stronger privacy requirements.

For both technologies, the efficiency drops linearly with the increased number of credentials being presented, and whenever inspectability, non-revocation proofs and inequality predicates are used. However, the number of disclosed attributes, number of attributes in a credential, and inequality predicates are relatively efficient. This knowledge is especially useful for system architects who need to understand the trade-off between using different features and the impact on the performance, e.g. defining some credentials as non-revocable in order to reduce the overhead on the presentation, whenever suitable.

Finally, we identified important issues that influence the trustworthiness of applications that use Privacy-ABC technologies, focusing on the trustworthiness of the Verifiers and Inspectors, and list important trust implications for potential deployments on the cloud.

Acknowledgment. The research leading to these results has received funding from the European Community's 7th Framework Programme under Grant Agreement no. 257782 for the project ABC4Trust, and from the Horizon 2020 research and innovation programme under grant agreement no. 653454 for the project CREDENTIAL.

We would also like to thank Welderufael B. Tesfay and Ahmed S. Yesuf for the fruitful discussions and the feedback that helped us improve the quality of the paper.

References

1. Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016). <http://data.consilium.europa.eu/doc/document/ST-5419-2016-INIT/en/pdf>
2. EU Commission. Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, OJ L 257, 28 August 2014, p. 73–114 (2012). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG
3. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. *Commun. ACM* **28**(10), 1030–1044 (1985)
4. Benenson, Z., Girard, A., Krontiris, I.: User acceptance factors for anonymous credentials: an empirical investigation. In: *Proceedings of the Workshop on the Economics of Information Security (WEIS)* (2015)

5. Microsoft research. U-Prove (2013) <http://research.microsoft.com/en-us/projects/u-prove/>
6. Bichsel, P., Binding, C., Camenisch, J., Gross, T., Heydt-Benjamin, T., Sommer, D., Zaverucha, G.: Cryptographic protocols of the identity mixer library. Technical report RZ 3730 (99740). IBM Research GmbH (2008)
7. Camenisch, J., Dubovitskaya, M., Lehmann, A., Neven, G., Paquin, C., Preiss, F.-S.: Concepts and languages for privacy-preserving attribute-based authentication. In: Fischer-Hübner, S., de Leeuw, E., Mitchell, C. (eds.) IDMAN 2013. IFIP AICT, vol. 396, pp. 34–52. Springer, Heidelberg (2013)
8. Bichsel, P., Camenisch, J., Groß, T., Shoup, V.: Anonymous credentials on a standard java card. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 600–610. ACM, New York (2009)
9. Sterckx, M., Gierlichs, B., Preneel, B., Verbauwhede, I.: Efficient implementation of anonymous credentials on java card smart cards. In: First IEEE International Workshop on Information Forensics and Security, WIFS 2009, pp. 106–110, December 2009
10. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials (extended version). IACR Crypt. ePrint Arch. **2010**, 496 (2010)
11. Camenisch, J.L., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)
12. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & #38; Communications Security, CCS 2013, pp. 1087–1098. ACM, New York (2013)
13. Chase, M., Zaverucha, G.: MAC schemes with efficient protocols and keyed-verification anonymous credentials (2013)
14. Batina, L., Hoepman, J.-H., Jacobs, B., Mostowski, W., Vullers, P.: Developing efficient blinded attribute certificates on smart cards via pairings. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 209–222. Springer, Heidelberg (2010)
15. Mostowski, W., Vullers, P.: Efficient U-Prove implementation for anonymous credentials on smart cards. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) SecureComm 2011. LNICST, vol. 96, pp. 243–260. Springer, Heidelberg (2012)
16. Vullers, P., Alpár, G.: Efficient selective disclosure on smart cards using Idemix. In: Fischer-Hübner, S., de Leeuw, E., Mitchell, C. (eds.) IDMAN 2013. IFIP AICT, vol. 396, pp. 53–67. Springer, Heidelberg (2013)
17. de la Piedra, A., Hoepman, J.-H., Vullers, P.: Towards a full-featured implementation of attribute based credentials on smart cards. In: Gritzalis, D., Kiayias, A., Askoxylakis, I. (eds.) CANS 2014. LNCS, vol. 8813, pp. 270–289. Springer, Heidelberg (2014)
18. Veseli, F., Serna-Olvera, J.: Benchmarking Privacy-ABC technologies - an evaluation of storage and communication efficiency. In: 2015 IEEE World Congress on Services, SERVICES 2015, New York City, NY, USA, 27 June–2 July 2015, pp. 198–205 (2015)
19. Paquin, C., Zaverucha, G.: U-Prove cryptographic specification v1.1 (revision 2). Technical report, Microsoft Corporation (2013)
20. Veseli, F., Vateva-Gurova, T., Krontiris, L., Rannenber, K., Suri, N.: Towards a framework for benchmarking Privacy-ABC technologies. In: Cuppens-Boulahia, N., Cuppens, F., Jajodia, S., El Kalam, A., Sans, T. (eds.) ICT Systems Security and Privacy Protection. IFIP Advances in Information and Communication Technology, vol. 428, pp. 197–204. Springer, Berlin Heidelberg (2014)

21. Stefan, A.: Brands: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)
22. Camenisch, J.L., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
23. Camenisch, J.L., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
24. Persiano, G., Visconti, I.: An efficient and usable multi-show non-transferable anonymous credential system. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 196–211. Springer, Heidelberg (2004)
25. European Commission. Horizon 2020 - work programme 2014–2015 - G. Technology Readiness Levels (TRL) (2014). https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf
26. Bichsel, P., Camenisch, J., Dubovitskaya, M., Enderlein, R., Krenn, S., Krontiris, I., Lehmann, A., Neven, G., Nielsen, J.D., Paquin, C., Preiss, F.-S., Rannenber, K., Sabouri, A., Stausholm, M.: D2.2 architecture for attribute-based credential technologies - final version. ABC4TRUST project deliverable (2014). <https://abc4trust.eu/index.php/pub>
27. ABC4Trust pilots. Abc4trust pilots (2015). <https://abc4trust.eu/index.php/home/pilots/>. Accessed 14 Dec 2015
28. Davis, F.D.: User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *Int. J. ManMachine Stud.* **38**, 475–487 (1993)
29. Jensen, J.L.: D4.4 smartphone feasibility analysis (2014). https://abc4trust.eu/download/Deliverable_D4.4.pdf
30. Blaze, M., Bleumer, G., Strauss, M.J.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998)
31. Smart, N. (ed.): D.SPA.20 ECRYPT II yearly report on algorithms and key sizes (2011–2012). Technical report, European Network of Excellence in Cryptology II, September 2012
32. Flavián, C., Guinalfú, M.: Consumer trust, perceived security and privacy policy: three basic elements of loyalty to a web site. *Ind. Manag. Data Syst.* **106**(5), 601–620 (2006)
33. Miriam, J.: Metzger: privacy, trust, and disclosure: exploring barriers to electronic commerce. *J. Comput.-Mediated Commun.* **9**(4) (2004)
34. Veseli, F., Sommer, D.M., Schallaboeck, J., Ioannis, K.: D8.12 architecture for standardization V2. ABC4TRUST project deliverable (2014). <https://abc4trust.eu/index.php/pub>

A Trust-Based Framework for Information Sharing Between Mobile Health Care Applications

Saghar Behrooz^(✉) and Stephen Marsh

Faculty of Business and Information Technology,
University of Ontario Institute of Technology, Oshawa, ON, Canada
{Saghar.Behrooz,Stephen.Marsh}@uoit.ca

Abstract. The use of information systems in the health care area, specifically in Mobile health care, can result in delivering high quality and efficient patient care. At the same time, using electronic systems for sharing information contributes to some challenges regarding privacy and access control. Despite the importance of this issue, there is a lack of frameworks in this area. In this paper, we propose a trust-based model for information sharing between mobile health care applications. This model consists of two parts, the first part calculates the needed amount of trust for sharing a specific part of information for each user, and the second part calculates the (contextual) current existing amount of trust. A decision for sharing information would be made based on a comparison between the components.

To examine the model, we provide different scenarios. Using mathematical analysis, we illustrate how the model works in those scenarios.

Keywords: Trust · Trust management · Mobile health care · Information security

1 Introduction

Development of technology has enabled mobile devices as appropriate tools for facilitating health care of various types, in what is known as M-health. M-health provides the opportunity to store and share the health information of a patient in their devices in order to deliver more efficient services, from fitness advice to more physician-oriented tools. However, security of the information in addition to access control is one of the controversial issues in this area [18]. Health care applications collect and share various type of information regarding physical activities and the lifestyles of users in addition to their medical and physiological information [16]. This is a privacy issue that could be better managed.

According to the National Committee for Vital and Health Statistics (NCVHS), health information privacy is “An individual’s right to control the

acquisition, uses, or disclosures of his or her identifiable health data. Confidentiality, which is closely related, refers to the obligations of those who receive information to respect the privacy interests of those to whom the data relate. Security is altogether different. It refers to physical, technological, or administrative safeguards or tools used to protect identifiable health data from unwarranted access or disclosure” [6].

In 2013 more than 35000 health apps existed for iOS and Android. From the most 600 useful ones, only 183 (30.5%) address mobile health privacy policies in some meaningful way [28]. Recently, both Google and Apple announced new platforms for health apps such as Health Kit [2], Research Kit and Google kit [5], which provide the possibility of information sharing between health care applications in one place.

We consider apple health kit as an example. Currently, each application is individually responsible for obtaining the trust of the user in order to get access to their health information. Health information has been divided into different categories. However, once the user shares their information, they have no further control over it. There has been extensive research in this area, however, this research has been focused on making policies or traditional mechanisms such as data encryption. Considering the importance of this information, in addition to usual privacy measurements, other considerations in design need to be met.

To address this, in our current research, we are proposing a trust model which aims to be used as leverage for the owner of the information to make decisions about sharing their information or part of it with a specific application. To formalize the trust value for each purpose of information, there are two components of a trust model which must be calculated. The first component of the model calculates the amount of trust which each person needs for sharing a specific part of the information. The second component of the model calculates the amount of trust already extant between a device and the application which is asking for the information.

The paper is organized as follows. In Sect. 2 we examine related work, before presenting our proposed model in Sect. 3, and a worked analysis in Sect. 4. We conclude with future work in Sect. 5.

2 Background

In this section, we review existing information systems in the health and M-health area. For the sake of brevity, we look at HealthKit, Apple’s health framework in detail. In addition, we provide a summary of current trust-based models and architectures in this area.

2.1 Healthcare Information Systems

Health Information Systems utilize data processing, information and knowledge in order to deliver quality and efficient patient care in health care environments

[12]. In recent years, there has been a great deal of movement towards computer-based systems from paper-based systems in health care environments [14]. Computer based systems provide the possibility of patient-centric systems instead of location constrained systems [7]. Furthermore, targeted users of these systems have also changed. Computer-based systems originally targeted only health care professionals, but gradually they have come to involve patients and their relatives as well [11]. Developments in these information systems over the previous decades provide the possibility of use of data for care planning and clinical research in addition to patient care purposes [11]. In addition, continuous health status monitoring using wearable devices such as sensors and smart watches further enhances the patient experience [17].

Expansions in use of data and health information in parallel with advancements in technology contributed to development of different architectures and information systems in this field. M-health is the use of mobile devices and their information in the health care area [26]. Special characteristics of mobile devices make them an excellent choice for this purpose. Their mobility and ability to access the information in addition to their ubiquity are some of these characteristics [26]. Employing technologies such as text messaging for tracking purposes, cameras for data collection, documentation and their ability to use cellular networks for internet connection, enable mobile devices to act as a perfect platform for delivery of health interventions [15]. Determining exact location through employing positioning technology, is also helpful for emergency situations [26] and device comfort purposes [19], where devices can determine how, when, and where to share relevant health information. Poket Doktor System (PDS) is one of the primary architectures in this area. This system includes an electronic patient device which contains electronic health care records, health care provider device and a communication link between them [29].

One of the major uses of mobile devices in health care is for monitoring purposes. Intelligent mobile health monitoring system (IMHMS) [25], introduces an architecture which is the combination of 3 main parts. Through a wearable body network, the system collects data and sends it to the patient's personal network. This network, based on the normal range of the index in question, logically decides whether to send the information to an intelligent medical server or not. The intelligent medical server is monitored by a specialist. Due to the broadness of the field, different monitoring systems have been introduced for specific purposes.

Some architectures have been introduced in order to improve the privacy of health care in this area. Weerasinghe et al. [30] present a security capsule with token management architecture in order to have secure transmission and data storage on device. Some models also use access control for healthcare systems based on users behaviours [31]. In [33] the authors propose a role-based prorogate framework. Some architectures have been developed in order to decrease clinical errors. For example, [32] proposes a scenario based diagnosis system which extracts relative clinical information from electronic health records based on the most probable diagnostic hypothesis.

2.2 Information Platform Example: Apple Healthkit

The HealthKit framework, which was introduced by Apple in iOS 8, lets health and fitness applications as well as smart devices gather health information about a user in one location. The framework provides services in order to share data between health and fitness applications. Through the HealthKit framework different applications can get access to each other's data with the user's permission. Users also can view, add, delete and manage data in addition to edit sharing permission using this app [1]. The framework can automatically save data from compatible Bluetooth LE heart rate monitors and the M7 motion coprocessor into the HealthKit store [3].

All the data which is managed by HealthKit is linked through the HealthKit store. Each application needs to use the HealthKit store in order to request and get permission for reading and sharing the health data [4].

Currently each application is individually responsible for obtaining the trust of the user in order to get access to their health information. The user has the control over the data and can decide whether to share data with the app or not. Users can also share some part of data whilst not giving permission for sharing another part [3].

In order to maintain the privacy of a user's data any application in the HealthKit must have a privacy policy. Personal health records models and HIPAA guidelines can be used in order to create these policies [3].

In addition, data from the HealthKit store cannot be sold. Data can be given to a third party app for medical research with owner consent. The use of data must be disclosed to the user by the application [1].

2.3 Trust in Information Systems

Trust plays an important role in human daily life. Trust can be studied from different perspectives, depending on the person who defines trust and the type of trust [20]. There is wide literature exploring in different fields such as evolutionary biology, sociology, social psychology, economics, history, philosophy and neurology.

The use of Trust Models in electronic healthcare can be classified into two groups: sharing information and electronic health records and monitoring patients. Becker, Moritz and Sewell introduced Cassandra, a trust management system that is flexible in the level of expressiveness of the language by selecting an appropriate constraint domain. Also, they present the results of a case study, a security policy for a national Electronic Health Record system, demonstrating that Cassandra is expressive enough for large-scale real-world applications with highly complex policy requirements. The paper concludes with identifying implementation steps including: building a prototype, testing the EHR policy in a more realistic setting, and producing web-based EHR user interfaces [8].

Considering the importance of security in wireless data communication, [9] reviews the characteristics of a secure system and proposes a trust evaluation

model. Data confidentiality, authentication, access control and privacy are examples of mentioned security issues. In this system nodes are representative of each component of system. A trust relationship between nodes has been evaluated to determine trustworthiness of each node. The main difference between this system and related works is that trust value of each node computed based on increased shaped functions such as exponential while others use linear functions. This leads to increase of past behaviour impact on trust [9]. In [21] the authors developed a trust-based algorithm for a messaging system. In this system, each node is assigned a trust value based on their behaviour. At same time, each message was divided to 4 parts and only nodes with the total trust value possible to read all parts of the messages.

3 Our Trust Model

In this section a trust model that considers both personal and environmental aspects is presented. This model aims to be used by the owner of the information to make decisions about sharing their health (or indeed, any) information, or part of it, with a specific application.

To formalize the trust value for each purpose of information, there are two components which must be calculated. The first component of the model calculates the amount of trust which each person needs for sharing a specific part of the information. The amount of trust that already exists between a device and the application which is asking for the information is calculated through the second component of the model. In the end, by comparing the two values, advise on sharing the information is made.

Table 1 summarizes the notations used in this chapter:

3.1 Personal Perspective

The personal perspective layer of the model will calculate the amount of trust that the user requires in order to share the information or a specific part of it. This layer is based on preferences of the owners of the information. To formalize the proposed system, this research considers a scenario in which a specific part of health information of a user has been requested by a specific application. Based on personal characteristics and experiences of persons, their behavior varies towards information sharing [13]. Stone and Stone [27] explored links between personality of individuals and information privacy issues. Gefen et al. [10] determined that personality has an impact on trust in virtual environments.

In order to determine the privacy preferences of each user, various factors should be considered and specific trust values need to be assigned. In the following sections, these factors and the methodology of assigning the trust values are presented.

Table 1. Explanation of notations

Symbols	Explanation
S	Sensitivity of information
C	Category of information agent
j	The index of information categories
n_c	Number of information categories
A	Application agent
i	The index of applications
P	Purpose of use of information
k	The index of user purposes
m_p	Number of usage purposes
T_d	Recency of information
C_0	Default Trust value for all of the categories
P_0	Default Trust value for all of the purposes
R	Rating of application agent
v	Representative of application rating
SN	Social network agent
u	Representative of number of mutual friends
I	Installer of the application agent
t	Representative of the installer of the application
TR	Threshold for information sharing
T	Trust value

Sensitivity of Information: Sensitivity of information might differ for individuals [22–24]. To facilitate the subjectivity of sensitivity of each piece of health information for users, we give the user the chance for decision making for each piece of information. The most significant factors which have an impact on calculation of the trust value are the following.

Category of Information: (See Table 2) Some health information can alter over its lifetime. In our model, we used the Apple health kit categories which falls into two main groups. The first group, “Characteristics data” refers to data which does not change over time such as gender, blood type and date of birth. The second group of data has been collected through the device and might change over time [1, 3].

In our model, C_j represents different categories of information. For each category of information, users would assign a comfort value for sharing each category of information. This value would be between $(-1, +1)$.

Table 2. Information categories

Characteristic data	Sample data
Biological sex	Vital signs
Blood type	Sleep analysis
Date of birth	Body measurements
Fitzpatrick skin type	Fitness
	Nutrition

Purpose of Use of Information: Different mobile applications use health information for various purposes. Considering existing applications in health care in parallel with the iOS health framework, the aim of use of information categorized to at least one of the several groups.

In our model we use A_i to represent these categories, thus, for each application depending on its purpose, A_i , would be an element of at least one of the following sets:

$$A_i \in P_k \tag{1}$$

in which:

$$k = \begin{cases} Research \\ PersonalMonitoring \\ PublicHealthMonitoring \\ CommercialUsage \\ GovernmentalUsage \end{cases}$$

Depending on the personality and priorities of the users, they might be interested in sharing information for each purpose. For each purpose again, users would assign a comfort value for sharing the information.

We use a matrix in order to represent and determine the relationships between various categories and purposes. In this $m_p \times n_c$ matrix, columns represent categories and rows represent purposes. Each element of the matrix is the minimum number of the assigned (by the user, but with some defaults) value for a specific purpose and category.

$$S_{j,k} = \min(C_j, P_k)$$

$$S_{m_p, n_c} = \begin{matrix} & C_1 & C_2 & \dots & C_{n_c} \\ \begin{matrix} P_1 \\ P_2 \\ \vdots \\ P_{m_p} \end{matrix} & \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,n_c} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n_c} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m_p,1} & s_{m_p,2} & \dots & s_{m_p,n_c} \end{bmatrix} \end{matrix} \tag{2}$$

Through this matrix, the system is able to choose a specific part of information for specific purpose, instead of omitting a whole category of information.

If the purpose of the application which is asking for the information is unclear, average of assigned values for all purposes could be used as a trust value.

$$\frac{1}{m_p} \sum_{i=1}^{m_p} P_k \tag{3}$$

At this point the trust value for a specific information item in a specific context (application) would be a function of the following variables:

T_d : Delay Time: This factor is added in order to improve the privacy of the user. Users can decide on sharing part(s) of their information after a specific delay. This may result in decrease in sensitivity of information for the user. Users have 3 options for sharing, representing different time periods before information is released. Depending on the user’s preference, T_d would be equal to:

$$T_d = \begin{cases} 1.5, \text{ if Share immediately} \\ 1, \text{ if Share after one week} \\ 0.5, \text{ if Share after one month} \end{cases} \tag{4}$$

Then:

$$S = f(C, P, T_d) = T_d \cdot \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{bmatrix} \tag{5}$$

3.2 Context Perspective

The second component of the model examines the environment of a user at the time of giving permission for sharing the information. This is used when calculating the amount of trust that exists at any time by considering the following:

- Default Trust to the applications in question
- The application’s reputation based on its current rating
- Common friends in social networks using the application
- The person who suggested installation of the application for example health care provider versus old friend

A higher amount of existing trust results in a lower threshold.

3.3 Default Trust to Each Category and Purpose

Since at the beginning there is no information on the applications, the average trust value which was assigned by the user would be calculated for the sensitivity matrix.

C_0 = Default Trust value for all of the categories

P_0 = Default Trust value for all of the purposes

$$S_{i_0, j_0} = \min\left(\frac{1}{n_c} \sum_{j=1}^{n_c} C_j, \frac{1}{m_p} \sum_{k=1}^{m_p} P_k\right) \quad (6)$$

then:

$$S_{m_p, n_c} = \begin{matrix} & C_1 & C_2 & \dots & C_{n_c} \\ \begin{matrix} P_1 \\ P_2 \\ \vdots \\ P_{m_p} \end{matrix} & \left[\begin{matrix} \min(C_0, P_0) & \min(C_0, P_0) & \dots & \min(C_0, P_0) \\ \min(C_0, P_0) & \min(C_0, P_0) & \dots & \min(C_0, P_0) \\ \vdots & \vdots & \ddots & \vdots \\ \min(C_0, P_0) & \min(C_0, P_0) & \dots & \min(C_0, P_0) \end{matrix} \right] \end{matrix} \quad (7)$$

3.4 Application Rating (Public Social)

R_v represents the rating score of the application in our model, for a specific online rating of v . Considering who is seeking for the application R_v would have one of the following values:

$$R_v = \begin{cases} 0.5, & \text{if } v = \text{more than average} \\ 1, & \text{if } v = \text{less than average} \\ 2, & \text{if } v = \text{negative} \end{cases} \quad (8)$$

3.5 Social Network Friends

Another factor which has impact on the threshold is the number of friends in their social network who are using the application. SN_u represents the number of mutual friends who are using the same application. Considering the number of friends in common SN_u would value one of the followings:

$$SN_u = \begin{cases} 0.5, & \text{if } u = \text{More than 5 friends} \\ 1.5, & \text{if } u = \text{Less than 5 friends} \\ 1, & \text{if } u = \text{No mutual friend} \end{cases} \quad (9)$$

3.6 Installer of the Application

In health care information systems, the relationship of the person who is asking for the information to the owner of information can have a crucial impact on the existing level of trust between them. Therefore, for example, if a person involved in the patient care suggests an application, the application in question is seen as potentially more reliable. In this model, 3 scenarios have been considered for installing an application. I_t represents the source suggesting the application. Considering who is seeking for the application I_t would value one of the following:

$$I_t = \begin{cases} 0.5, & \text{if } t = \text{Healthcare provider suggests} \\ 0.75, & \text{if } t = \text{Proposed by a sensor the user already uses} \\ 1, & \text{if } t = \text{Randomly downloaded application} \end{cases} \quad (10)$$

3.7 Estimation of the Threshold

Considering all the factors the second component of the model would be:

$$TR = S_0 \cdot R_v \cdot SN_u \cdot I_t \quad (11)$$

Information would be shared if:

$$TR < T \quad (12)$$

4 Analysis

In this part, we examine our model using different scenarios as use case examples. Furthermore, different user personalities and various applications have been considered.

4.1 Various Agents

Personality and characteristics of people have a crucial impact on their decision making. In order to make allowances for this, in this experiment we divide the user agents to three main categories: optimistic, pessimistic and realistic. In the following section each category is described:

Optimist. An optimist believes in the best outcome in all the situations and expects the best results in everything [20]. In our examples, an optimist always selects the maximum trust value.

Pessimist. In the eyes of pessimist, in opposite to the optimist, the worst possible result is being seen. The pessimist expects the worst outcome in any situation. Therefore, the pessimist agent selects the worst trust value in all the situations [20].

Realist. However, in reality most people are some place between the two extremes. This situation also applies to agents. For the sake of simplicity in this paper, we randomly choose from intervals within the 4 quartiles in the spectrum from optimist to pessimist

4.2 Pool of Applications

In healthcare environments, various applications with different characteristics exist. This section looks at examples of these applications.

Application α . α has the following characteristics:

- It needs to have access to nutrition information, fitness information and vital signs.
- It uses information for commercial purposes, research purposes and also personal health monitoring.
- It has been rated less than average.

Application β . Application β has the following characteristics:

- This app needs access to sleep analysis information and nutrition information.
- It uses information for research purposes, personal health monitoring and public health.
- It has been rated higher than average.

4.3 Various Situations

Although personality plays a significant role in decision making other factors including the experiences of the user or their current mental state can affect their judgment. To address this, we test the model in 2 different scenarios.

Scenario 1 – Installing Random Applications. Tracy was browsing health care applications on the app store. One of the diet applications interested her and she installed it on her device. She did not have any past information about this application, no one has suggested it and none of her friends is using this application. This application needs to have access to her fitness information, nutrition information and weight information.

Scenario 2 – Various Rated Applications. Steve is a tech savvy person. He reads reviews of applications and downloads many health apps onto his device. Rating of the applications is the most effective reason for him to decide to download the application or not. Furthermore, he is willing to share his information for research purposes or for monitoring his own health. However, Steve is not interested in sharing for commercial uses. Recently, he has sleeping problems. In order to monitor himself he decides to install an sleep analysis application on his device.

4.4 Mathematical Analysis

In order to examine how the model works, in this part we briefly analyse the model in different situations.

Table 3. Trust values assigned by Tracy

Information category	Trust value	Purpose	Trust value
Vital signs	0.81	Research	0.22
Sleep analysis	0.32	Personal monitoring	0.31
Body measurements	0.46	Public health	0.46
Fitness	0.22	Commercial Usage	0.51
Nutrition	0.33	Governmental usage	0.73

Example 1. In the first scenario, we consider Tracy as an optimist. Therefore, she relatively assigns a higher trust value for sharing information. Table 3 represents the trust values she assigned for each purpose and category.

In the sensitivity matrix we have the minimum amount between each category and purpose, therefore:

$$S = f(C, P) = \begin{bmatrix} 0.22 & 0.22 & 0.22 & 0.22 & 0.22 \\ 0.31 & 0.31 & 0.31 & 0.22 & 0.31 \\ 0.46 & 0.32 & 0.46 & 0.22 & 0.33 \\ 0.51 & 0.32 & 0.46 & 0.22 & 0.33 \\ 0.73 & 0.32 & 0.46 & 0.22 & 0.33 \end{bmatrix} \tag{13}$$

$$t_d = 1.5 \tag{14}$$

Then the trust matrix would be:

$$S_{m,n} = \begin{matrix} & \begin{matrix} VitalSigns & SleepAnalysis & Dobymeasurements & Fitness & Nutrition \end{matrix} \\ \begin{matrix} Research \\ PersonalMonitoring \\ PublicHealth \\ CommercialUsage \\ GovernmentalUsage \end{matrix} & \begin{bmatrix} 0.33 & 0.33 & 0.33 & 0.33 & 0.33 \\ 0.46 & 0.46 & 0.46 & 0.33 & 0.46 \\ 0.69 & 0.48 & 0.69 & 0.33 & 0.49 \\ 0.76 & 0.48 & 0.69 & 0.33 & 0.49 \\ 1.09 & 0.48 & 0.69 & 0.33 & 0.49 \end{bmatrix} \end{matrix} \tag{15}$$

We consider that application α is the application which Tracy has downloaded. Therefore, we have:

$$S_{i_0,j_0} = \min\left(\frac{1}{5} \sum_{i=1}^5 C_j, \frac{1}{5} \sum_{i=1}^5 P_k\right) = \min(0.428, 0.444) = 0.428 \tag{16}$$

$$S = f(C, P) = \begin{bmatrix} 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \end{bmatrix} \tag{17}$$

And:

$$R_v = 1 \tag{18}$$

$$SN_u = 1 \tag{19}$$

$$I_t = 1 \tag{20}$$

The threshold matrix would be:

$$TR = \begin{bmatrix} 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \\ 0.428 & 0.428 & 0.428 & 0.428 & 0.428 \end{bmatrix} \tag{21}$$

Specific parts of information for specific purposes are expected to be shared if the value of corresponding member of sensitivity matrix is higher than the value of corresponding member in the threshold matrix. Therefore, fitness information wont be shared since the trust value is less than the threshold. However, nutrition information and vital signs information will be shared since for the purpose in which application α using those information, trust value is higher than the threshold.

$$0.33 < 0.428 \rightarrow \textit{Do not share} \tag{22}$$

$$0.46 > 0.428 \rightarrow \textit{Share vital signs information for personal monitoring} \tag{23}$$

Example 2. In the second scenario, we considered Steve as a pessimist. He does not give high trust values to the application. Therefore he assigns the following trust values as noted in Table 4.

Table 4. Trust values assigned by Steve

Information category	Trust value	Purpose	Trust value
Vital signs	-0.31	Research	-0.22
Sleep analysis	-0.68	Personal monitoring	0.11
Body measurements	0.23	Public health	-0.47
Fitness	-0.46	Commercial Usage	-0.86
Nutrition	-0.33	Governmental usage	-0.59

In the sensitivity matrix we have the minimum amount between each category and purpose, therefore:

$$S = f(C, P) = \begin{bmatrix} -0.31 & -0.68 & -0.22 & -0.46 & -0.33 \\ -0.31 & -0.68 & 0.11 & -0.46 & -0.33 \\ -0.47 & -0.68 & -0.47 & -0.47 & -0.47 \\ -0.86 & -0.86 & -0.86 & -0.86 & -0.86 \\ -0.59 & -0.68 & -0.59 & -0.59 & -0.59 \end{bmatrix} \tag{24}$$

Steve decides to share his information after one week.

$$t_d = 1 \tag{25}$$

Then the trust matrix would be:

$$S_{m,n} = \begin{matrix} & \begin{matrix} VitalSigns & SleepAnalysis & Bodymeasurements & Fitness & Nutrition \end{matrix} \\ \begin{matrix} Research \\ PersonalMonitoring \\ PublicHealth \\ CommercialUsage \\ GovernmentalUsage \end{matrix} & \begin{bmatrix} -0.31 & -0.68 & -0.22 & -0.46 & -0.33 \\ -0.31 & -0.68 & 0.11 & -0.46 & -0.33 \\ -0.47 & -0.68 & -0.47 & -0.47 & -0.47 \\ -0.86 & -0.86 & -0.86 & -0.86 & -0.86 \\ -0.59 & -0.68 & -0.59 & -0.59 & -0.59 \end{bmatrix} \end{matrix} \tag{26}$$

We consider that application β is the application which Steve has installed. Therefore, we have:

$$S_{i_0,j_0} = \min\left(\frac{1}{5} \sum_{i=1}^5 C_j, \frac{1}{5} \sum_{i=1}^5 P_k\right) = \min(-0.31, -0.406) = -0.406 \tag{27}$$

$$S = f(C, P) = \begin{bmatrix} -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \end{bmatrix} \tag{28}$$

And:

$$R_v = 1 \tag{29}$$

$$SN_u = 1 \tag{30}$$

$$I_t = 1 \tag{31}$$

The threshold matrix would be:

$$TR = \begin{bmatrix} -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \\ -0.406 & -0.406 & -0.406 & -0.406 & -0.406 \end{bmatrix} \tag{32}$$

Again, by comparing matrix elements, a recommended decision for information sharing can be made. In this case, sleep analysis information won't be shared. Also, nutrition information wont be shared as application β use this information for public health purposes.

$$-0.68 < -0.406 \rightarrow \textit{Do not share} \tag{33}$$

$$-0.33 > -0.406 \rightarrow \textit{Share nutrition information for research} \tag{34}$$

$$-0.33 > -0.406 \rightarrow \textit{Share nutrition information for personal monitoring} \tag{35}$$

$$-0.47 < -0.406 \rightarrow \textit{Do not share nutrition information} \tag{36}$$

5 Conclusions and Further Work

In this paper, we proposed a trust model which calculates the required trust value of information sharing between health care mobile applications, in addition to the existing amount of trust. By employing a trust model, we believe we can be proactive and prevent sharing parts of the information which put the privacy of the user in danger. Moreover, by categorizing the information and purpose of use, we aim to provide the opportunity for sharing in different levels. Going forward, we plan to implement the framework and the corresponding user interfaces, and a user study is in the planning stage.

Acknowledgment. The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada under the Discovery Program.

References

1. Apple inc. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/HealthKit.html>
2. Apple inc. health kit. <https://developer.apple.com/healthkit>
3. Apple inc, the health kit framework. https://developer.apple.com/library/ios/documentation/HealthKit/Reference/HealthKit_Framework/
4. Apple inc, the health kit framework. https://developer.apple.com/library/ios/documentation/HealthKit/Reference/HKHealthStore_Class/index.html#/apple_ref/occ/cl/HKHealthStore
5. Google. <https://developers.google.com/fit/?hl=en>
6. National committee on vital and health statistics. privacy and confidentiality in the nationwide health information network, June 2006. <http://www.ncvhs.hhs.gov/060622lt.html>
7. Ball, M.J., Lillis, J.: E-health: transforming the physician/patient relationship. *Int. J. Med. Inform.* **61**(1), 1–10 (2001)
8. Becker, M.Y., Sewell, P.: Cassandra: flexible trust management, applied to electronic health records. In: *Computer Security Foundations Workshop, 2004. Proceedings. 17th IEEE*, pp. 139–154. IEEE (2004)
9. Boukerche, A., Ren, Y.: A secure mobile healthcare system using trust-based multicast scheme. *IEEE J. Sel. Areas Commun.* **27**(4), 387–399 (2009)
10. Gefen, D., Benbasat, I., Pavlou, P.: A research agenda for trust in online environments. *J. Manage. Inf. Syst.* **24**(4), 275–286 (2008)
11. Haux, R.: Health information systems-past, present, future. *Int. J. Med. Inform.* **75**(3), 268–281 (2006)
12. Haux, R., Winter, A., Ammenwerth, E., Brigl, B.: *Strategic Information Management in Hospitals: An Introduction to Hospital Information Systems*. Springer Science Business Media, New York (2013)
13. Hsu, M.H., Ju, T.L., Yen, C.H., Chang, C.M.: Knowledge sharing behavior in virtual communities: the relationship between trust, self-efficacy, and outcome expectations. *Int. J. Hum. Comput. Stud.* **65**(2), 153–169 (2007)
14. Jydstrup, R.A., Gross, M.J.: Cost of information handling in hospitals. *Health Serv. Res.* **1**(3), 235 (1966)

15. Klasnja, P., Pratt, W.: Healthcare in the pocket: mapping the space of mobile-phone health interventions. *J. Biomed. Inform.* **45**(1), 184–198 (2012)
16. Kotz, D., Avancha, S., Baxi, A.: A privacy framework for mobile health and home-care systems. In: *Proceedings of the First ACM Workshop on Security and Privacy in Medical and Home-care Systems*, pp. 1–12. ACM (2009)
17. Lukowicz, P., Kirstein, T., Troster, G.: Wearable systems for health care applications. *Methods of Information in Medicine-Methodik der Information in der Medizin* **43**(3), 232–238 (2004)
18. Mandl, K.D., Markwell, D., MacDonald, R., Szolovits, P., Kohane, I.S.: Public standards and patients' control: how to keep electronic medical records accessible but privatemedical information: access and privacydoctrines for developing electronic medical recordsdesirable characteristics of electronic medical recordschallenges and limitations for electronic medical recordsconclusionscommentary: Open approaches to electronic patient recordscommentary: A patient's viewpoint. *BMJ* **322**(7281), 283–287 (2001)
19. Marsh, S., Wang, Y., Noël, S., Robart, L., Stewart, J.: Device comfort for mobile health information accessibility. In: *2013 Eleventh Annual International Conference on Privacy, Security and Trust (PST)*, pp. 377–380. IEEE (2013)
20. Marsh, S.P.: *Formalising Trust as a computational concept*. Ph.D. thesis
21. Narula, P., Dhurandher, S.K., Misra, S., Woungang, I.: Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing. *Comput. Commun.* **31**(4), 760–769 (2008)
22. Nowak, G.J., Phelps, J.: Understanding privacy concerns. An assessment of consumers' information-related knowledge and beliefs. *J. Direct Mark.* **6**(4), 28–39 (1992)
23. Phelps, J., Nowak, G., Ferrell, E.: Privacy concerns and consumer willingness to provide personal information. *J. Public Policy Mark.* **19**(1), 27–41 (2000)
24. Podsakoff, P.M., MacKenzie, S.B., Lee, J.Y., Podsakoff, N.P.: Common method biases in behavioral research: a critical review of the literature and recommended remedies. *J. Appl. Psychol.* **88**(5), 879 (2003)
25. Shahriyar, R., Bari, M.F., Kundu, G., Ahamed, S.I., Akbar, M.M.: Intelligent mobile health monitoring system (imhms). *Int. J. Control Autom.* **2**(3), 13–28 (2009)
26. Siau, K., Shen, Z.: Mobile healthcare informatics. *Inform. Health Soc. Care* **31**(2), 89–99 (2006)
27. Stone, E.F., Stone, D.L.: Privacy in organizations: theoretical issues, research findings, and protection mechanisms. *Res. Pers. Hum. Res. Manage.* **8**(3), 349–411 (1990)
28. Sunyaev, A., Dehling, T., Taylor, P.L., Mandl, K.D.: Availability and quality of mobile health app privacy policies. *J. Am. Med. Inform. Assoc.* **22**(e1), e28–e33 (2015)
29. Vawdrey, D.K., Hall, E.S., Knutson, C.D., Archibald, J.K.: A self-adapting healthcare information infrastructure using mobile computing devices. In: *5th International Workshop on Enterprise Networking and Computing in Healthcare Industry, Healthcom 2003. Proceedings*, pp. 91–97. IEEE (2003)
30. Weerasinghe, D., Rajarajan, M., Rakocevic, V.: Device data protection in mobile healthcare applications. In: Weerasinghe, D. (ed.) *eHealth 2008. LNICST*, vol. 1, pp. 82–89. Springer, Heidelberg (2009)
31. Yarmand, M.H., Sartipi, K., Down, D.G.: Behavior-based access control for distributed healthcare environment. In: *21st IEEE International Symposium on Computer-Based Medical Systems, CBMS 2008*, pp. 126–131. IEEE (2008)

32. Yousefi, A., Mastouri, N., Sartipi, K.: Scenario-oriented information extraction from electronic health records. In: 22nd IEEE International Symposium on Computer-Based Medical Systems, CBMS 2009, pp. 1–5. IEEE (2009)
33. Zhang, L., Ahn, G.J., Chu, B.T.: A role-based delegation framework for healthcare information systems. In: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, pp. 125–134. ACM (2002)

Supporting Coordinated Maintenance of System Trustworthiness and User Trust at Runtime

Torsten Bandyszak¹(✉), Micha Moffie², Abigail Goldsteen²,
Panos Melas³, Bassem I. Nasser³, Costas Kalogiros⁴, Gabriele Barni⁵,
Sandro Hartenstein⁶, Giorgos Giotis⁷, and Thorsten Weyer¹

¹ paluno – The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Essen, Germany
{torsten.bandyszak, thorsten.weyer}@paluno.uni-due.de

² IBM Research, Haifa, Israel
{moffie, abigailt}@il.ibm.com

³ IT Innovation Centre, University of Southampton, Southampton, UK
{pm, bmn}@it-innovation.soton.ac.uk

⁴ Athens University of Economics and Business, Athens, Greece
ckalog@aueb.gr

⁵ D-CIS Lab, Thales R&T, Delft, The Netherlands
Gabriele.Barni@D-CIS.NL

⁶ Department of Economics, Brandenburg University of Applied Sciences,
Brandenburg an der Havel, Germany
sandro.hartenstein@fh-brandenburg.de

⁷ Athens Technology Center S.A., Athens, Greece
g.giotis@atc.gr

Abstract. In addition to design-time considerations, user trust and the trustworthiness of software-intensive socio-technical systems (STS) need to be maintained during runtime. Especially trust can only be monitored based on the actual usage of the system in operation. Service providers should be able to make informed decisions about runtime adaptation based on trust and trustworthiness, as well as respective essential relations. In this paper we present a unified approach to support the coordination of trust and trustworthiness maintenance. Trustworthiness maintenance is based on measuring objective system qualities, while trust maintenance considers two complementary measures of trust, i.e., the user behavior, and an estimation of the perceived system trustworthiness. A prototype tool demonstrates the feasibility of our approach. Furthermore, we illustrate specific functionalities of the tool by means of an application example.

Keywords: Trust · Trustworthiness · Run-Time maintenance

1 Introduction

The success of software-intensive socio-technical systems (STS) increasingly depends on their users' trust in relevant system properties determining the system's trustworthiness. We consider trustworthiness of an STS its ability to fulfill the stakeholders'

expectations (cf. [1]), which depends on a multitude of measureable software quality attributes, such as reliability or security [2, 3]. In contrast, trust is a relationship between a person (trustor) and a system (cf. [1]). Trust involves subjectivity and uncertainty as it can be seen as a guess that the software will perform as expected (cf. [1]).

In order to assure the trustworthy operation of an STS and a high level of user trust in the system, it is not sufficient to consider these aspects during development. In particular, a user's trust can only be measured when the system is actually in operation. Service providers should thus be able to systematically assess both trust and trustworthiness at runtime, which requires early planning and installation of suitable monitoring sensors as well as actuators to invoke mitigations (cf. [4]). Furthermore, there is a reasonable interrelation between trust and trustworthiness, which only becomes visible at runtime. Besides the fact that trust is related to system trustworthiness, however, other factors influence the trust relationship as well. For instance, due to user experience, lack of transparency of the system's behavior, or the provider's reputation (cf. e.g. [1, 5]), the current subjective trust in the system may be low although objective system quality properties actually indicate a high degree of trustworthiness. Such a mismatch of trust and trustworthiness should be avoided [5]. Especially in complex STS, users may influence each other's trust as well. Hence, for making well-informed decisions on mitigation actions, it is crucial to consider comprehensive runtime information about the levels of both trust and trustworthiness.

Regarding trustworthiness maintenance, there are several approaches towards monitoring the quality of services (e.g. [6, 7]). These approaches often only consider a few quality properties to be monitored, e.g., response time or availability. On the other hand, trust measurement and management approaches (e.g., [8, 9]) deal with measuring the user behavior, or determining the user-perceived quality of specific services, most notably for real-time internet communication. However, these isolated approaches fail to address the challenges described above, i.e., to support making runtime decisions taking both trust and trustworthiness as well as the relationship between them into account. Hence, there is a lack of approaches that combine trust and trustworthiness aspects for runtime monitoring and management.

In this paper, we present our unified approach combining the maintenance of system trustworthiness and users' trust in STS. Our approach supports service administrators in coordinating the monitoring and management of trust and trustworthiness at runtime, as well as performing related mitigations. Trustworthiness maintenance is based on monitoring software services within an STS, and identifying threats caused by software properties (i.e., qualities or functions) not fulfilling the user expectations. We extend our preliminary results [10, 11] with respect to trustworthiness maintenance, in order to consider trust and reflect the essential relationship between trust and trustworthiness as motivated above. Regarding trust maintenance, our approach combines two complementing ways of measuring users' trust: (1) estimating trust based on the different users' perceptions of the system's trustworthiness characteristics, and (2) monitoring the trust-related user behavior (e.g., in terms of number of mouse clicks in a certain time frame), which is heavily dependent on the type of STS.

Furthermore, we describe a tool prototype that implements our runtime trust and trustworthiness maintenance approach, and demonstrates its technical feasibility. The tool also allows validating our approach, and eliciting new requirements for extensions.

As an initial evaluation of our approach, we applied the prototype to a case example of a secure web chat system. This application example involves three evaluation scenarios, each focusing on specific aspects of trust and trustworthiness maintenance. Details on the tool and the application example can be found in [12].

The remainder of this paper is organized as follows: Sect. 2 discusses related work. Section 3 provides an overview of our approach and sketches the conceptual solution. Section 4 presents a tool prototype that supports the application of our approach. In Sect. 5, we describe an application example that provides initial results of the ongoing evaluation of our approach. Finally, Sect. 6 concludes the paper.

2 Related Work

Regarding runtime monitoring and measurement of trustworthiness, an initial overview of related work can be found in our previous work [10]. Approaches for online monitoring of software services, e.g. [6], are based on observing the quality level of a service (QoS) with respect to guarantees defined in SLAs. These approaches usually cover only a limited set of quality properties such as availability or response time (cf., e.g. [7]). An overview of tools for monitoring QoS of cloud services is given in [13].

In [14], different service composition constructs and cost are taken into account for evaluating and managing the trustworthiness of a composite service-based system. A combination of quality properties monitoring and reputation is also possible (e.g. [7, 14]). The framework and trustworthiness evaluation method presented by Lenzi et al. [15] supports managing trust relationships, and aims at evaluating the trustworthiness of a trusted component with respect to the satisfaction of quality attributes and the expectation that these will remain satisfied and stable. For managing trustworthiness at runtime, the system composition can be adjusted, e.g., underperforming services or components may be substituted or restarted (cf., e.g. [15]). QoS-aware service selection also takes cost minimization objectives into account (cf., e.g. [16]).

Compared to user surveys, measuring the user behavior directly from the interactions with the system is a more promising approach for runtime trust maintenance. It is, however, challenging to define generic trust-related behavioral measures and metrics that can be used for runtime monitoring. Leichtenstern et al. [17] investigated the physiological behavior of website users by means of heart rate and eye tracking sensors to determine how to objectively measure trust-related behavior (attention and engagement). Regarding security, Blindspotter [18] is a user behavior monitor that aims at detecting abnormal user activities caused by e.g. hijacked accounts.

As mentioned in the introduction, the trust of users is also related to the perceived trustworthiness of the system, e.g., in terms of response times. In general, transparency of the system's trustworthiness characteristics helps achieve appropriate trust [5]. Studies such as [8, 9] indicate a relation between the subjective quality of experience (QoE), i.e., the "overall acceptability of an application or service, as perceived subjectively by the end-user" [19], and the objective QoS, e.g., the user-perceived throughput on network level. This is particularly relevant for browsing and real-time web applications such as online gaming or VoIP (cf., e.g. [20]). A framework for measuring QoE of video conferencing, and controlling QoE in case of limited

bandwidth is presented in [21]. For instance, QoE Monitor [22] and EvalVid [23] are free tools that support determining the perceived video conference transmission quality. Fiedler et al. [24] propose a generic formula to quantify relationships between QoE and network-level QoS, which aims at controlling QoE based on QoS monitoring.

For managing user trust, acceptable QoS characteristics of a system, e.g., the latency of web browsing, should be determined to allow for appropriate resource allocation [9]. To control QoE, additional parameters need to be considered as well. Zhang and Ansari [25] propose a framework for managing QoE that distinguishes a QoS/QoE reporting and a QoE management component to satisfy users' target QoS constraints.

As motivated in Sect. 1, evaluating the trustworthiness of a system together with the trust of its users is of vital importance for runtime maintenance. Some research effort has been spent to analyze the relationship between objective trustworthiness and subjective user trust. However, our state of the art analysis revealed a gap of approaches considering the combined runtime evaluation of both trust and trustworthiness to provide administrators comprehensive information for coordinating adaptation decisions. Furthermore, regarding complex, heterogeneous STS, there is a lack of approaches covering versatile quality characteristics, and different, complementary means to assess user trust. Available tools are also mostly based on narrow measurement concepts, or focus on specific applications. Sections 3 and 4 will introduce our conceptual solution and our prototype tool addressing these particular research gaps.

3 Coordinated Trust and Trustworthiness Maintenance

Figure 1 gives an overview of our overall conceptual approach, which supports administrators in coordinating runtime maintenance of trustworthiness and trust within an STS. The monitor observes the behavior of the STS constituents (including software, humans, etc.), and reports respective misbehavior alerts to the management, which then determines appropriate controls to be executed by the mitigation. Since the mitigation is rather a technical issue, this paper focuses on monitoring and management.

To monitor trustworthiness at runtime, related functional and quality properties, such as response times, are reported by STS-specific sensors in the form of events. Based on these events, trustworthiness metrics are calculated to identify violations of user expectations in terms of the demanded quality of the software, which may be specified in SLAs for relevant quality criteria. Violations of specified values indicate the presence of threats (cf. [26]), which keep the system from performing as expected.

Concerning the monitoring of user trust in an STS, our approach subsumes the following two complementary approaches to support trust management decisions:

- The current level and evolution of a user's trust can be estimated at runtime based on the user's perception of the system's trustworthiness, which is characterized by respective metrics. This is based on the premise that each user is classified into one of four groups or segments, according to expected trust-related behaviors and relevant trustor attributes. Identifying the segment a user belongs to is done by

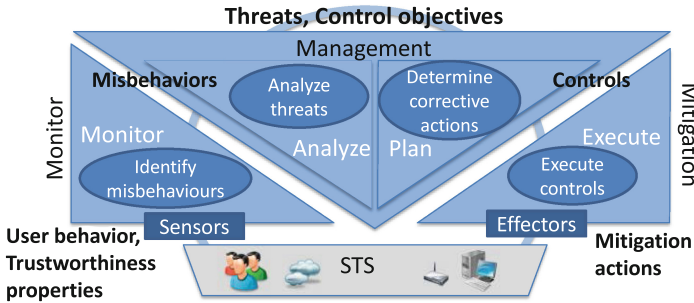


Fig. 1. Overview of our approach (based on [10])

contrasting the answers to be given in a predefined questionnaire before using the system to those already collected during a training period. An initial trust value and the update coefficients have been calculated for each of the four segments, and validated based on survey research. For example, members of the ‘High Trust’ group were found to consistently overestimate trustworthiness. For more details on user segmentation and trust level computation please see [27]. Calculating trust levels and selecting corresponding controls is based on the assumption that users use the system for a certain period (called optimization window). Based on their trust level at the end of each period, they decide whether to keep using the system, or not.

- Trust is also monitored and maintained by analyzing the user-specific, trust-related behavior. This approach considers each user separately, and requires respective STS sensors to report user-specific behavioral information. For instance, the number of a user’s questions raised in a certain time interval can be considered a valuable source of trust-related information (see application example in Sect. 5).

Similar to trustworthiness maintenance (cf. [10, 11]), both of these trust monitoring approaches are used to identify threats that are related to user trust. If a decrease in trust is detected based on either the estimated trust level indicating the user experience of trustworthiness, or abnormal user behavior when interacting with the system, an alert is issued to trigger the trust management process to analyze potential threats. Respective thresholds are defined for each of the user segments (cf. [27]). The management then analyses the likelihood of threats activity using semantic reasoning.

In case of any active threat to trust or trustworthiness, suitable controls are identified and selected based on a cost/benefit analysis (see [28] for more details on selecting an optimal control). The controls are then applied by executing mitigation actions on the STS. A control could be applied automatically (e.g., shutting down or substituting an underperforming service), or chosen and carried out by the administrator (e.g., contacting a specific user). Applying controls to restore trustworthiness will also reflect in the trust levels of the users. However, the relation between trust and trustworthiness also depends on other factors, which may, for instance, only be visible through monitoring the user behavior. By considering both trust and trustworthiness, our unified approach supports administrators in identifying and coordinating reasonable relationships between trust and trustworthiness, and their evolution during runtime.

4 Tool Prototype

Based on [12], this section presents the tool prototype supporting our approach. It will present the tool’s architecture as well as the major components, and the user interface.

4.1 Tool Architecture

The initial tool architecture was presented in [10], including the overall tasks of the three main components Monitor, Management and Mitigation. This architecture was further refined in [11], describing the components involved in maintaining trustworthiness (TW). Figure 2 shows the final tool architecture, including new components for trust (T) monitoring and management, as well as an optimal control selector, and a user interface for configuring, managing and viewing the trust and trustworthiness status of the system. The subsequent sections will describe the main components.

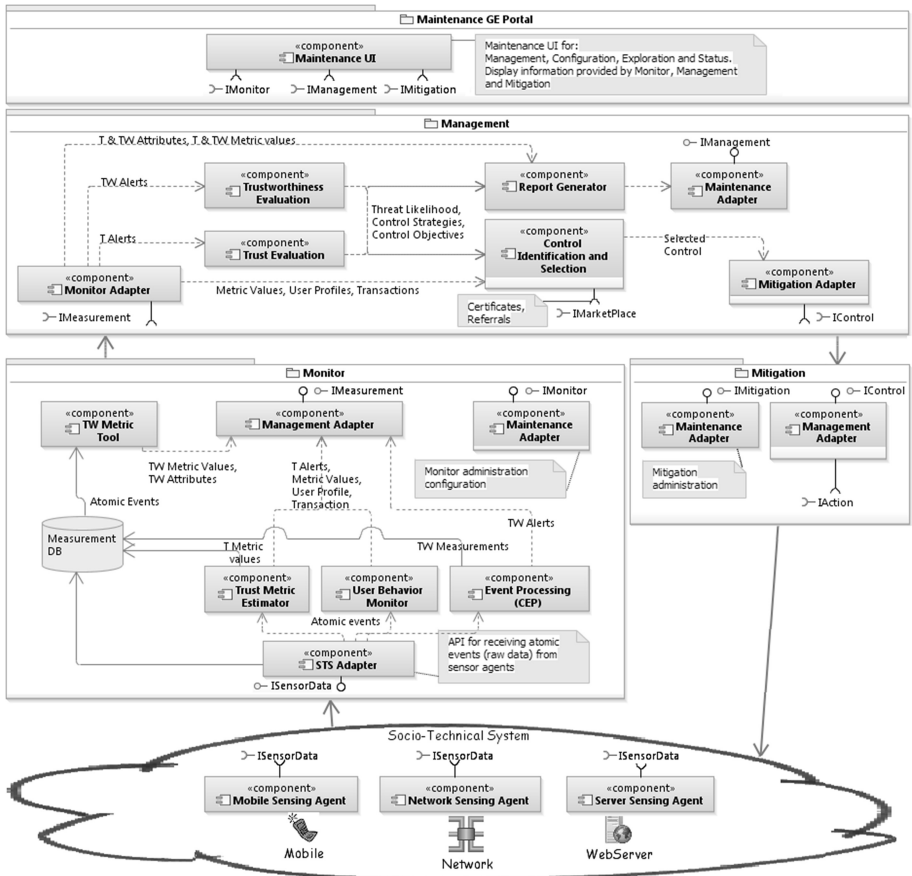


Fig. 2. Architecture of the trust and trustworthiness maintenance tool [12]

The *Monitor* is connected to the STS via system-specific sensors that report trust- and trustworthiness-relevant runtime monitoring data (cf. [4]). The *STS Adapter* forwards all atomic events received from the STS to three components in charge of the initial processing of these events and generating alerts upon deviation from normal (TW) or desired (T) behavior. The *Complex Event Processing* (CEP) is in charge of system-wide trustworthiness-related events, the *Trust Metric Estimator* (TME) estimates system trustworthiness as perceived by its different users, and the *User Behaviour Monitor* (UBM) collects data from user-specific sensors and estimates the trust each user has in the system. The *Monitor* also saves all atomic events in a *Measurement Database*, to be used in the *GUI Maintenance Portal* or for generating reports.

Both trustworthiness and trust alerts generated in the *Monitor* are forwarded to the *Management* component, where trustworthiness alerts are further processed by the *Trustworthiness Evaluator* (TWE), and trust alerts are further processed by the *Trust Evaluator* (TE). Both of these components generate a list of potential threats (including their likelihoods), and possible control strategies and objectives for mitigating these threats. The *Control Identification and Selection* component, using the *Optimal Control Selector* (OCS) sub-component, then suggests the most cost-effective control to be selected and deployed by the *Mitigation*. Respective feedback on the deployment of a control (i.e., the execution of a concrete mitigation action) is fed back to the respective *Management* components so that they can update their internal state accordingly. All detected threats, control strategies and deployed controls are also saved in a database to be used later in the *Maintenance Portal* UI or for generating reports.

4.2 Monitor Components

Complex Event Processing (CEP). The CEP detects misbehaviors of the STS, indicating potential threats to trustworthiness. It handles atomic events reported by STS-specific sensors, which are needed for monitoring trustworthiness. In different contexts (e.g., time intervals), these incoming events are aggregated to perform an initial analysis. To this end, a pre-defined configuration involves rules to detect patterns of related events. Based on the incoming sensor events and STS-specific detection rules, alerts are issued to the *Management* components for further threat analysis.

Trustworthiness Metric Tool. The Trustworthiness Metric Tool component serves as a repository for managing metrics to measure the trustworthiness of STS constituents, and estimate the user perception of trustworthiness. It allows browsing system quality attributes contributing to trustworthiness, as well as defining metrics details such as computation formulas. The tool also supports computing metrics at runtime, with reported trustworthiness properties as inputs. Metric values can be retrieved by other components, e.g., by the UI to generate reports covering longer time periods.

Trust Metric Estimator (TME). The TME is a Bayesian computational model that aims at estimating a user's trust level over time for a number of metrics defined by the *Trustworthiness Metric Tool*, e.g., trust with respect to reliability. These trust levels are calculated based on the personality of each trustor (retrieved from respective attributes,

such as competency level and trust stance, stored in the *Customer Profile DB*), and system trustworthiness properties. The TME receives atomic trustworthiness-related events from the STS. In particular, a successful transaction increases trust and vice versa, while the magnitude of trust change depends on the user's segment (see [27] for more details on the trust computation, and the four segments that were found to have statistically significant differences). Figure 3 shows the TME design.

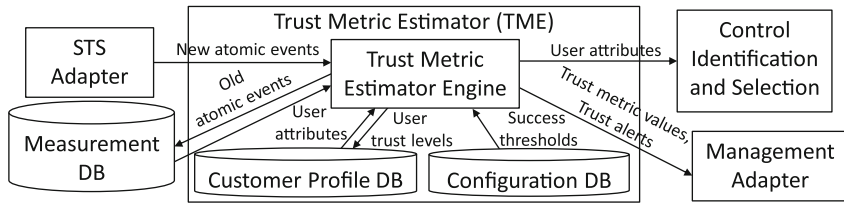


Fig. 3. Conceptual design of the Trust Metric Estimator

A particular transaction performed by a user is characterized as successful, or not, by comparing the atomic event value with the threshold value defined by the administrator for that metric, and eventually is stored in the *Measurement DB* (cf. Fig. 2). To this end, versatile trustworthiness characteristics (such as response time etc.) can be chosen. If the current trust level of any user in the system exceeds the thresholds set by the administrator in the *Configuration DB*, a trust alert is generated by the *TME Engine* and forwarded to the *Management*. The evolution of trust levels is also stored in the *Customer Profile DB*, and provided to the *Maintenance Portal* and the *OCS*.

User Behaviour Monitor (UBM). The purpose of the UBM is to continuously monitor and measure trust-related behaviors of individual users through respective sensors indicating these behaviors. It relates the behaviors to trust disposition of the user and to a model of trust in the system. Although the UBM supports any kind of STS, the sensors are system-specific and need to be configured accordingly. Each sensor needs to be configured with the following three parameters:

- A *low threshold* allows raising alerts in case the trust level estimated by a sensor drops under this limit, indicating a potential low trust perceived by the user.
- A *high threshold* is used for raising alerts in case the trust level exceeds the defined value, suggesting that the system is over-performing.
- Additionally, a *weight* is set, which is used to evaluate the overall trust level.

The UBM collects trust-related atomic events from different sources, stores these events in a persistent database, and performs an initial processing to aggregate them and compute metric values characterizing trust in terms of user behavior. When a certain trust measurement reaches a predefined threshold, which can be either too low or too high trust, a respective alert is generated and forwarded to the *Management*, which will then select an appropriate mitigation control to restore trust in the use of the system. The UBM consists of three main modules, as shown in Fig. 4:

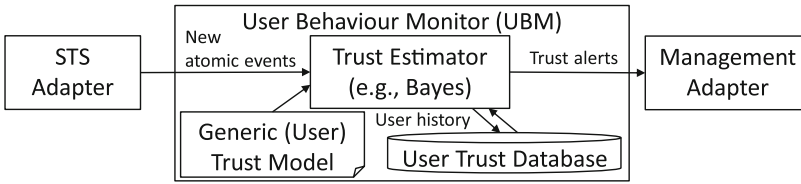


Fig. 4. Conceptual design of the User Behaviour Monitor

- The *Generic (User) Trust Model* comprises a list of application-specific sensors, which are configured with the corresponding parameters mentioned above.
- The *Trust Estimator* processes input from the sensors, analyses the data, and consecutively issues alerts in the case of a trust violation.
- In the *User Trust Database* the trust history for every user is stored.

4.3 Management Components

Trustworthiness Evaluation (TWE). The TWE is responsible for identifying and classifying threats related to trustworthiness based on calculating threat likelihoods, as well as for determining appropriate control objectives. Threat and control identification is based on machine reasoning using a generic threat ontology that incorporates relevant knowledge, and an internal runtime model of all the different STS constituents and their behaviors. This model is incrementally updated at runtime. Background on the models used for trustworthiness evaluation can be found e.g. in [11]. Figure 5 shows a simplified conceptual design of the TWE, comprising four main modules:

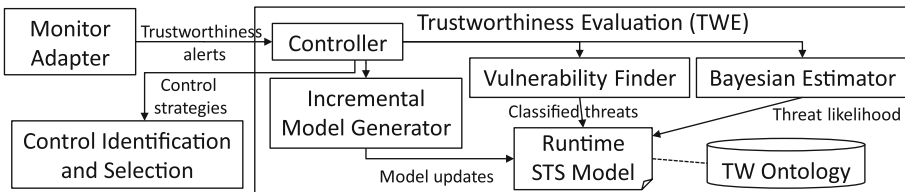


Fig. 5. Conceptual design of the Trustworthiness Evaluation

- A *Controller* handles incoming requests and maps them to the responsible module.
- The *Incremental Model Generator* incrementally updates the *Runtime STS Model* according to events reported by the *Monitor* (i.e., the CEP). Runtime event handling mechanisms are used to also reflect system topology updates (e.g., considering the deployment of controls) and changes of control objectives in real time.
- The *Vulnerability Finder* enforces control rules as defined in the *Threat Ontology* to classify trustworthiness threats into blocked or mitigated threats, secondary effects, or vulnerabilities, according to the presence of controls.

- The *Bayesian Estimator* analyses the likelihood of all threat activity given the reported trustworthiness behaviors of the STS. This quantitative threat analysis is based on a well-defined statistical model using Bayesian networks.

Trust Evaluation (TE). The TE receives alerts from the *Monitor* components TME and UBM, and analyses them by means of an internal, system-specific trust model to detect the current threats that may arise due to changes in user trust. Based on the threat evaluation, the TE proposes control strategies for the successive mitigation actions. To this end, an interface is used by the *Control Identification and Selection* to find active trust threats (querying the current runtime models) and propose a list of control strategies. The TE is composed of the following main modules (see Fig. 6):

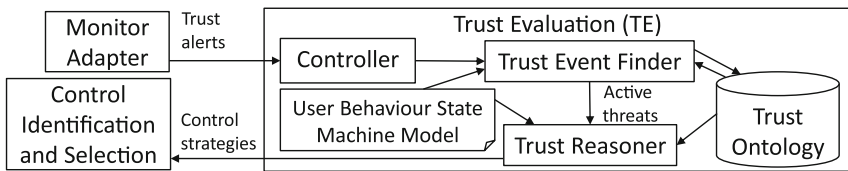


Fig. 6. Conceptual design of the Trust Evaluation

- The *Trust Evaluation Controller* preprocesses the trust alerts from the *Monitor* to determine their relative types and the priority to handle them.
- The *Trust Event Finder* discovers trust-related threats based on the trust alerts. To this end, an application-specific representation of each user's expected behavior, i.e., the *User Behaviour State Machine Model*, is utilized. A misbehavior may indicate the presence of a threat due to a lack of user trust.
- The *Trust Reasoner* determines how to handle the threats discovered by the *Event Finder*. It proposes a list of applicable control strategies for subsequent mitigation.
- The *Trust Ontology* database keeps track of the application-specific trust terms that are used in the *User Behaviour State Machine Model*.

Optimal Control Selector (OCS). The OCS (a subcomponent of the *Control Identification and Selection*, see Fig. 7) suggests the most cost-effective control(s) to be deployed in order to deal with a threat regarding a trustworthiness misbehavior, and/or user trust concern. More specifically, it maximizes the probability that the metric, for which an alert was triggered, will have an acceptable value after a certain number of transactions, while keeping the expected costs low. A provider can maximize its expected profits using the approach below (see [28] for more details):

1. Estimate the initial trust level of all users in the service and for the particular metric associated to the incoming alert from the TE. These alerts are retrieved from the TME via the *Monitor Adapter*. This has to be performed for each user segment, rather than for each individual user.
2. Compute the minimum of successes necessary for each user segment to reach the initial trust level after a number of transactions (i.e., the optimization window).

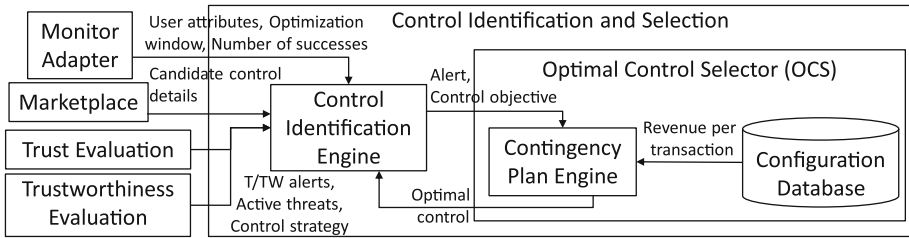


Fig. 7. Conceptual design of the Optimal Control Identification and Selection

3. The *Contingency Plan Engine* creates the so-called contingency plan for reaching the initial trust level in a cost-effective way by solving a dynamic programming problem and identifies the current optimal control. In order to do so, we need details on candidate controls (i.e., price and trustworthiness properties) from a marketplace providing alternative services that can be deployed as possible controls.

4.4 Design of the Maintenance Portal User Interface

The *Maintenance Portal* shall provide the administrator information on the current and past state of both trust and trustworthiness, enabling her to detect any trust and trustworthiness violation, understand the root cause of the violation, and, finally, to approve and/or choose the most appropriate mitigation. In order to address these needs, the UI has been designed and implemented with the following capabilities:

1. Display all the necessary graphs to reproduce current and past trends of all the different trust and trustworthiness parameters relevant for runtime maintenance.
2. Provide the user with runtime trust and trustworthiness information on different levels of abstraction, starting from a general view of the overall trust and trustworthiness behaviors to more detailed graphs illustrating, e.g., trust of different users, or lower-level trustworthiness properties.
3. Notify the administrators anytime, no matter which page is actually displayed, of any relevant event about the status of the system (e.g., alerts and threats).
4. Visualize detailed information about these raised alerts and the detected threats, following any reported trust and trustworthiness violation.
5. Propose a list of applicable controls in order to mitigate the detected threats.
6. Allow the administrator to select and apply one of the proposed mitigation actions.

Figure 8 shows the UI landing page. It shows the overall system trust and trustworthiness levels. This screen also shows the alerts to be handled. The administrator can browse more detailed levels, e.g., depicting the trustworthiness per quality attribute or a specific constituent of the STS. To avoid frequent pop-ups, the user is notified of new alerts using an icon on the top right (bell icon); only when the user clicks it, the complete alert information will be displayed. The notification table enables the user to

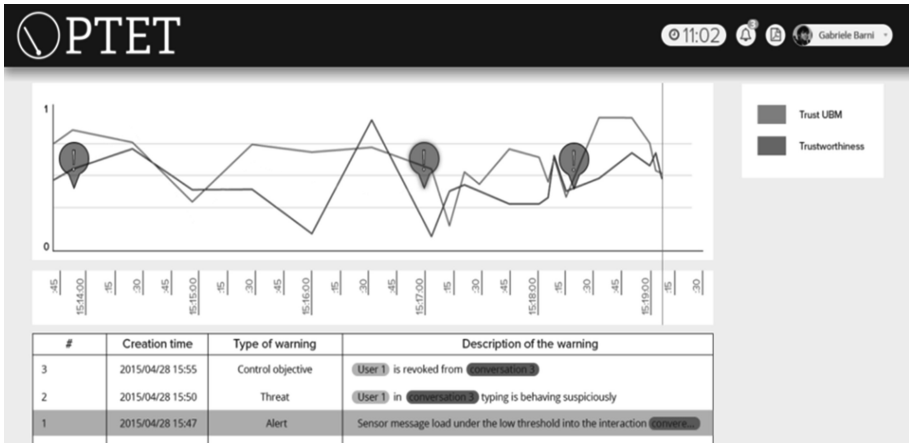


Fig. 8. Main screen of the Maintenance Portal user interface

immediately take action¹ by pressing the ‘Take Action’ button in case the situation requires mitigation. More details on each of the UI screens can be found in [12].

In addition to the feedback received via the UI, runtime reports (i.e., XML documents) can also be generated to provide offline feedback to system administrators on the STS’ trust and trustworthiness status during different time intervals, or to be consumed by other tools, such as an online software marketplace.

5 Application Example

The application example for demonstrating the trust and trustworthiness maintenance tool is based on a Secure Web Chat (SWC) system. The SWC addresses the need for trustworthy online communication, which may be vital in case of a cyber-attack, so that users of critical STS can ask for advice, and administrators can discuss appropriate actions or consult external experts. SWC users can create or join secure chat rooms to discuss critical topics and exchange files. Hence, the SWC is a complex STS consisting of software and hardware infrastructure, but also a multitude of human users. The SWC mainly faces the following trust and trustworthiness concerns:

- A high level of user trust: Although the usage of the SWC may be mandatory in case there is no other means for secure communication, a user with low trust in the SWC may not be able to contribute adequately to solving a cyber crisis.
- Accurate real-time communication: The chat room participants need to communicate efficiently in order to manage a cyber crisis. Furthermore, low performance or failures of the system will cause users mistrusting the SWC.

¹ Note that, based on configuration, the tool may select mitigation actions automatically, or query the administrator.

As mentioned, our initial evaluation aims at demonstrating and verifying our prototype tool and thereby showing the feasibility of our approach, as well as allowing further validation. We designed three evaluation scenarios to systematically focus on specific aspects of trust and/or trustworthiness maintenance, and the corresponding features of our prototype. Based on the SWC, we specified simplified input data (i.e., events reported from SWC-specific sensors), and determined the expected output of the tool components. We simulated the event stream to exercise specific, functionally related components and thereby invoke the involved functionalities separately.

In [29] we already briefly sketched our evaluation plan involving the three scenarios, which will be described in the following. In particular, we will explain the behavior, the use, as well as the responses and outcomes of the different tool components for illustration purposes. The results of the exemplary application show that the tool performs as expected, and sustain our confident that the tool will be useful in practice.

Trust Scenario. This scenario focuses on user trust in terms of trust-related behavior (see Sect. 3). Hence, it specifically evaluates the UBM and the TE. To monitor the evolution of user trust over time, the users have been segmented, and initial trust levels have been computed based on our trust computation approach sketched in [27].

Regarding the user behavior, we configured suitable UBM sensors reporting respective events. Based on the SWC use case, we made some assumptions for detecting trust-related user behaviors, in order to simplify the complex matter. In practice, trust monitoring will demand for more elaborate concepts (cf. [5]). Our approach allows for defining system-specific sensors and thereby tailoring or refining the user behavior monitoring to a specific system to be monitored. In our example, unusual user behaviors focus on the message activity of each user. Abnormal user behaviors indicate a lack of trust in the SWC and the other chat participants. Hence, the evaluation scenario involves the following two applications of trust maintenance:

- A SWC user in a chatroom raises many questions, and sends many messages. The input events reported by SWC-specific sensors carry the numbers of messages in general and questions in particular, which occurred in a given time period for each user. These events are evaluated against configured thresholds. For instance, in case a user raises more than five questions in a reporting period of two minutes, the UBM calculates a decrease in the current trust level. In contrast, e.g. less than three questions will reflect in an increase in the trust level.
- A SWC user participates very slowly, and mainly contributes very short messages. Based on respective system-specific monitoring data and thresholds, the UBM computes an increase or decrease in trust for all the monitored users in a chatroom. For instance, messages shorter than 30 characters will cause a decrease in trust.

The UBM continuously updates each user's trust level. Based on thresholds defined for the trustor segments (cf. [27]), it issues alerts to the TE, which updates the internal model, identifies low trust threats, and proposes suitable controls. In both situations described above, the following mitigations are proposed to the SWC administrator:

- Automatically notify the user having low trust of either unclearness regarding the user's motives, or about a check of the situation.

- Open a special communication channel (e.g., a separate chat room) allowing the administrator to discuss and solve the user's trust misbehavior.
- Exclude the abnormally behaving chat user from the conversation.

Trustworthiness Scenario. For service providers, it is essential to ensure the QoS at a level satisfactory for the service consumers, such as the SWC users. As mentioned, the SWC is faced with reliability and robustness concerns, as well as high availability. This scenario shows the following trustworthiness maintenance applications:

- Reliability and robustness are measured based on the SWC's ability to handle exceptions. Any occurring exception is reported by an atomic event that also indicates whether the exception was successfully handled. The CEP aggregates these events to calculate the ratio of recovered exceptions and the total number of exceptions. If this ratio is too low, the CEP reports an alert to the TWE. The TWE continuously updates its internal runtime model to reason about current threats, and asserts a software malfunction threat. Finally, the Mitigation interacts with the TWE to determine software patching as the control objective to mitigate the threat.
- To measure the availability of the SWC, the CEP aggregates "alive" notifications reported from suitable SWC sensors using pings. A simple detection rule is used to compare the current mean availability against a predefined threshold (e.g., 95 %), and to alert the TWE about underperforming SWC services. To counteract the related threat of an under-provisioned service, scalability is the suggested control objective to counteract the threat, e.g., by load balancing or adding resources.

Optimal Control Selection Scenario This scenario demonstrates the identification of the most cost-effective control to mitigate a threat, in case there is a set of suitable controls available. It focuses on the second aspect of our trust maintenance approach (see Sect. 3), i.e., it deals with monitoring user trust based on the perceived level of trustworthiness. Hence, threats pertaining to user trust are mapped to control strategies affecting the trustworthiness of the system by restoring the system's QoS.

In this scenario, we consider trustworthiness (and the effect of its perception on user trust) in terms of the response times of the SWC for processing and delivering chat messages. Similar to the trust scenario, in a first monitoring step the trust levels of the users (which are grouped into four trustor segments) is continuously updated. Thresholds are defined in the TME, so that it can issue low trust alerts to the TE.

The TE then identifies a control strategy, i.e., a set of potential controls to be considered for mitigating the threat. In this application example, we defined three different options for restoring trustworthiness by substituting underperforming services. For each service that can serve as a substitute, its trustworthiness (in terms of response time), and the associated cost are defined. A trusted software marketplace provides the relevant metric values for each of these controls so that they can be compared. The possible controls are passed to the OCS component, which then identifies the optimal control. The suggested control may change over time, when the trust-decreasing effect of high response times is active over a longer period.

6 Conclusion and Future Work

In this paper, we described a unified approach complementing runtime trust and trustworthiness maintenance, and a corresponding tool prototype. Our unified approach specifically addresses the challenge of relating and coordinating objective system trustworthiness and subjective user trust at runtime by presenting system administrators comprehensive information about both, and thereby supporting the decision-making process for maintaining complex STS. Our trustworthiness maintenance is based on observing measurable system properties that contribute to the trustworthiness of the STS, while the trust maintenance relies on quantifying the subjective user trust through monitoring user behavior and estimating the perception of system trustworthiness characteristics. The tool prototype demonstrates the technical feasibility, and allows further investigating the validity of our approach. An initial evaluation illustrates a potential application of the tool, and shows that the different prototype components work as expected in different scenarios involving different functionality.

Future work should focus on a more elaborate validation of our approach. To this end, the tool prototype could be applied to a real industry case example in order to further evaluate the benefits of our approach and discuss it with potential stakeholders. This will contrast using our approach with using existing tools. Furthermore, general interdependencies between trust and trustworthiness can be examined using the tool. The resulting information may be used to discover potential for extensions or refinements of our approach, and, ultimately, to define concepts and techniques for balancing trust and trustworthiness at runtime.

Acknowledgements. This work was supported and funded by the EU FP7 project OPTET (grant no. 317631).

References

1. Miller, K.W., Voas, J.: The metaphysics of software trust. *IT Prof.* **11**(2), 52–55 (2009)
2. Hasselbring, W., Reussner, R.: Toward trustworthy software systems. *IEEE Comput.* **39**(4), 91–92 (2006)
3. Mohammadi, N.G., Paulus, S., Bishr, M., Metzger, A., Könnecke, H., Hartenstein, S., Weyer, T., Pohl, K.: Trustworthiness attributes and metrics for engineering trusted internet-based software systems. In: Helfert, M., Desprez, F., Ferguson, D., Leymann, F. (eds.) *CLOSER 2013. CCIS*, vol. 453, pp. 19–35. Springer, Heidelberg (2014)
4. Bandyszak, T., Gol Mohammadi, N., Bishr, M., Goldsteen, A., Moffie, M., Nasser, B.I., Hartenstein, S., Meichanetzoglou, S.: Cyber-physical systems design for runtime trustworthiness maintenance supported by tools. In: *REFSQ Workshops*, pp. 148–155. CEUR (2015)
5. Muir, B.M.: Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics* **37**(11), 1905–1922 (1994)
6. Clark, K.P., Warnier, M.E., Quillinan, T.B., Brazier, F.M.T.: Secure monitoring of service level agreements. In: *5th International Conference on Availability, Reliability, and Security (ARES)*, pp. 454–461. IEEE (2010)

7. Zhao, S., Wu, G., Li, Y., Yu, K.: A framework for trustworthy web service management. In: 2nd International Symposium on Electronic Commerce and Security, pp. 479–482. IEEE (2009)
8. Shaikh, J., Fiedler, M., Collange, D.: Quality of experience from user and network perspectives. *Ann. Telecommun.* **65**(1), 47–57 (2010)
9. Bouch, A., Bhatti, N., Kuchinsky, A.J.: Quality is in the eye of the beholder: meeting users' requirements for internet quality of service. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), pp. 297–304. ACM (2000)
10. Gol Mohammadi, N., Bandyszak, T., Moffie, M., Chen, X., Weyer, T., Kalogiros, C., Nasser, B., Surridge, M.: Maintaining trustworthiness of socio-technical systems at run-time. In: Eckert, C., Katsikas, S.K., Pernul, G. (eds.) TrustBus 2014. LNCS, vol. 8647, pp. 1–12. Springer, Heidelberg (2014)
11. Goldsteen, A., Moffie, M., Bandyszak, T., Gol Mohammadi, N., Chen, X., Meichanetzoglou, S., Ioannidis, S., Chatzidiam, P.: A tool for monitoring and maintaining system trustworthiness at runtime. In: REFSQ Workshops, pp. 142–147. CEUR (2015)
12. OPTET Consortium: D6.4.2 – Measurement and Management Tools (2nd release). Technical report, http://www.optet.eu/wp-content/uploads/deliverables/OPTET_WP6_D6.4.2_Measurement_and_Management_tools_2nd_Release_V2.0.pdf
13. Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F., Prakash Jayaraman, P., Khan, S.U., Guabtni, A., Bhatnagar, V.: An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. *Computing* **97**(4), 357–377 (2015)
14. Elshaafi, H., McGibney, J., Botvich, D.: Trustworthiness monitoring and prediction of composite services. In: 2012 IEEE Symposium on Computers and Communications, pp. 000580–000587. IEEE (2012)
15. Lenzini, G., Tokmakoff, A., Muskens, J.: Managing trustworthiness in component-based embedded systems. *Electron. Notes Theoret. Comput. Sci.* **179**, 143–155 (2007)
16. Yu, T., Zhang, Y., Lin, K.: Efficient algorithms for web services selection with end-to-end QoS constraints. *ACM Trans. Web* **1**(1), 1–26 (2007)
17. Leichtenstern, K., Bee, N., André, E., Berkmüller, U., Wagner, J.: Physiological measurement of trust-related behavior in trust-neutral and trust-critical situations. In: Wakeman, I., Gudes, E., Jensen, C.D., Crampton, J. (eds.) Trust Management V. IFIP AICT, vol. 358, pp. 165–172. Springer, Heidelberg (2011)
18. BalaBit IT Security Blindspotter. <https://www.balabit.com/network-security/blindspotter>
19. ITU: Vocabulary and Effects of Transmission Parameters on Customer Opinion of Transmission Quality, Amendment 2, ITU-T Rec. P.10/G.100. ITU (2006)
20. Huang, T.-Y., Chen, K.-T., Huang, P., Lei, C.-L.: A generalizable methodology for quantifying user satisfaction. *IEICE Trans. Comm.* **E91-B**(5), 1260–1268 (2008)
21. Vakili, A., Grégoire, J.-C.: QoE management for video conferencing applications. *Comput. Netw.* **57**, 1726–1738 (2013)
22. Saladino, D., Paganelli, A., Casoni, M.: A tool for multimedia quality assessment in NS3: QoE monitor. *Simul. Model. Pract. Theory* **32**, 30–41 (2013)
23. Klaue, J., Rathke, B., Wolisz, A.: EvalVid – A framework for video transmission and quality evaluation. In: Kemper, P., Sanders, W.H. (eds.) TOOLS 2003. LNCS, vol. 2794, pp. 255–272. Springer, Heidelberg (2003)
24. Fiedler, M., Hossfeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. *IEEE Netw.* **24**(2), 36–41 (2010)
25. Zhang, J., Ansari, N.: On assuring end-to-end QoE in next generation networks: challenges and a possible solution. *IEEE Commun. Mag.* **49**(7), 185–192 (2011)
26. Surridge, M., Nasser, B., Chen, X., Chakravarthy, A., Melas, P.: Run-time risk management in adaptive ICT systems. In: Proceedings of ARES 2013, pp. 102–110. IEEE (2013)

27. Kanakakis, M., van der Graaf, S., Kalogiros, C., Vanobberghen, W.: Computing trust levels based on user's personality and observed system trustworthiness. In: Conti, M., Schunter, M., Askoxylakis, I. (eds.) TRUST 2015. LNCS, vol. 9229, pp. 71–87. Springer, Heidelberg (2015)
28. Kalogiros, C., Kanakakis, M., van der Graaf, S., Vanobberghen, W.: Profit-maximizing trustworthiness level of composite systems. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2015. LNCS, vol. 9190, pp. 357–368. Springer, Heidelberg (2015)
29. Bishr, M., Heinz, C., Bandyszak, T., Moffie, M., Goldsteen, A., Chen, W., Weyer, T., Ioannidis, S., Kalogiros, C.: Trust and trustworthiness maintenance - from architecture to evaluation. In: Conti, M., Schunter, M., Askoxylakis, I. (eds.) TRUST 2015, LNCS 9229, pp. 319–320. Springer, Heidelberg (2015)

Limitations on Robust Ratings and Predictions

Tim Muller^(✉), Yang Liu, and Jie Zhang

Nanyang Technological University, Singapore, Singapore
t.j.c.muller@gmail.com

Abstract. Predictions are a well-studied form of ratings. Their objective nature allows a rigorous analysis. A problem is that there are attacks on prediction systems and rating systems. These attacks decrease the usefulness of the predictions. Attackers may ignore the incentives in the system, so we may not rely on these to protect ourselves. The user must block attackers, ideally before the attackers introduce too much misinformation. We formally axiomatically define robustness as the property that no rater can introduce too much misinformation. We formally prove that notions of robustness come at the expense of other desirable properties, such as the lack of bias or effectiveness. We also show that there do exist trade-offs between the different properties, allowing a prediction system with limited robustness, limited bias and limited effectiveness.

1 Introduction

Ratings are an important tool in online cooperation. Ratings are used in, e.g., recommender systems, trust and reputation systems, e-commerce systems and security systems [10–12]. We reason about a specific type of predictions, namely those that we can judge in hindsight – called predictions. Prediction are also an interesting topic of research in themselves [1]. Typically, users that give predictions that are better (accurate or honest) are rewarded by becoming more credible. However, there are incentives outside of the system that may drive a user to give worse (inaccurate or dishonest) predictions. These unfair ratings attacks are well-known in literature, and found to occur in real systems. On a robust prediction system, the impact of these unfair ratings is limited.

A standard technique in prediction systems is to have a mechanism to encourage users to behave in a certain way, by setting the right incentives. However, in practice, users may have a bigger incentive to give bad predictions. We know that users attack systems by providing false predictions [9, 14] despite losing credit within the system. A user that ignores the incentives of a system is called an attacker. In other words, an attacker does not necessarily care about the rewards and punishments that the prediction system sets, because incentives outside of the system (e.g. bribes, collusion) are greater.

As we cannot modify the behaviour of these attackers, we must resort to interpreting the predictions in a robust fashion. Specifically, we must somehow limit the impact of unfair ratings. In this paper, we introduce notions of robustness (differing in strength) that codify that the amount of noise that a single

agent can introduce is limited. We have a threefold motivation for the exact formulations: intuitive grounds, information theory and hypothesis testing. Given our definition of robustness, we can prove a specific prediction system to be robust.

Robustness comes at a cost. With no tolerance towards misinformation, any useful way of using predictions is impossible. For weaker robustness requirements, we have more subtle impossibilities regarding the use of predictions. One main contribution in this paper is a general and rigorous proof that robustness, bias and effectiveness are tradeoffs, and that certain combinations are impossible. The proofs are axiomatic, meaning that we have axioms for the various levels of robustness, bias and effectiveness, and we prove that no model can satisfy all of them. Specifically:

- No meaningful model exists for absolute robustness (no tolerance towards misinformation).
- Any model for strict robustness (fixed misinformation threshold) has some bias and a finite lifespan.
- Any model for weak robustness (growing tolerance towards misinformation) has some bias and cannot be fully effective.

The results are summarised in Table 1.

Fortunately, if we are willing to make the trade-offs, then robust models do exist. We show that a prediction system with strict robustness can exist and be useful despite its hampered effectiveness. Similarly, we also show that a prediction system with weak robustness can be implemented and be far more effective. These results extend only to prediction systems, since they rely on the user knowing the validity of the predictions after the fact.

The paper is organised as follows. First we discuss work related to our domain, but also impossibility results in social choice which inspired the methodology. In Sect. 3, we present the requirements that we want a prediction system to fulfill, in natural language. Then, in Sect. 4, we present a formal model of predictions, events, filters and misinformation. In Sect. 5, we formalise the requirements in that model. In Sect. 6, we establish the relationships between the axioms – particularly we close the bridge between the information-theoretic and the statistical perspective on the quality of predictions. In Sect. 7, we prove the impossibility results. All the limitations of robust prediction systems can be found in this section. In Sect. 8, we prove the existence of prediction systems that have robustness, albeit with considerable reduction of effectiveness. In these latter two sections, non-technical formulations of the results are presented in bold font. We provide a conclusion in Sect. 9.

2 Related Work

Our original research interest lies in robust ratings, as e.g. in [16, 17]. There, the ratings are quantified using information theory. This idea is not novel, as e.g. [11] also uses information theory to quantify ratings. Our novelty is that we

Table 1. Effectiveness given levels of robustness (**AR**, **SR**, **WR**), bias (**SU**, **WU**) and non-prescience (**T**).

	WU	SU	WU & T	SU & T	T
AR	0	0	0	0	0
SR (θ)	2^θ	0	θ	0	θ
WR (f)	$> 2^{f(1)}$	0	$f(n)$	0	$f(n)$

are able to formulate a system with a strict robustness cutoff. The damage of attacks is strictly limited. However, for the system to work, the ratings must be predictions – verifiable in hindsight.

Prediction systems are widely used and studied [3, 8]. An important type of prediction system is a prediction market. There has been lots of research on prediction markets, especially their resistance against manipulation [4, 6, 9]. An inherent problem of prediction markets is that raters insensitive to the system’s incentives have absolute freedom to manipulate [4, 6]. Our approach limits the influence of individual raters, without taking away the ability to predict.

We formulate a set of axioms that we want prediction systems to satisfy. That approach is inspired on social choice theory [15]. Arrow’s impossibility Theorem [2] states that the result of a vote must (1) have X over Y if all prefer X over Y , (2) let the order of X and Y be independent of Z and (3) there is no dictator. In fact, robustness against manipulation is also a well-studied issue there [15]. Our axioms are fundamentally different, but the idea that certain combinations of axioms do not admit a model is directly taken from social choice theory.

3 Axiomatic Requirements

A trust system is robust, when it operates well under attacks. A common way to increase the robustness of the system, is to try to detect attacks. While such detection mechanisms certainly mitigate attacks, they cannot prevent them, by their nature. A detection mechanism detects attacks that have already occurred (at least partially). Ideally, however, we can prevent the attacks from occurring in the first place.

In this paper, we are concerned with attacks that introduce misinformation to the users. However, first, not all attacks induce noise towards the user, but break the system in other ways (e.g. the reputation lag attack [13], where the attacker exploits a time delay before his infamy is spread)¹. Thus, we only consider attacks by strategic unfair predictions. Second, not all unfair prediction attacks are harmful (e.g. the camouflage attack, where users are honest to gain trust, and betray others when trusted; see [17]). We ignore attacks that do not introduce noise to the user; attacks that do not (aim to) deceive users. Rather, we look at gathering predictions in a robust manner.

¹ We ignore security attacks, such as identity theft or denial of service attacks.

The requirements in this section are informally defined using natural language. Later, we formally define our terminology, and translate the requirements to formal statements. For the sake of precision, we fix the meaning of some terms: A *prediction* is a statement about an *event* before it occurs. An event has an *outcome*, after which the degree of correctness of the prediction is known. *Noise* is the inverse of that degree of correctness. A *rater* is an agent (human or system) that produces predictions. Ratings are *accurate* when they assign the true (subjective) probabilities to outcomes (i.e. outcomes assigned $x\%$ happen $x\%$ of the time), and raters are accurate when they produce accurate ratings. See Sect. 4 for a more formal definition of the terminology.

3.1 Robustness Requirements

Consider the strictest formulation of robustness, called absolute robustness (**AR**): **No rater may introduce noise**. Note that **AR** implies that no group of raters may introduce noise either. Intuitively, **AR** seems too strong. If no predictions can introduce any noise, no matter how small or improbable, then how can the rater make any meaningful predictions? In fact, in Sect. 7, we prove this intuition correct; no non-trivial system can be absolutely robust.

The generalisation of absolute robustness is θ -strict robustness (**SR**): **No rater may introduce noise larger than θ** . Note that **SR** implies that no group of raters sized n may introduce noise larger than $\theta \cdot n$. Strict robustness is also a strong axiom, and it is somewhat workable, although it negatively affects other aspects of the system. Particularly, due to the fixed-size tolerance of noise, and the inevitability of noise, any rater can only provide a limited number of predictions. We refer to this property as *effectiveness*, and its axiom is stated below.

Strict robustness can be weakened, e.g. by allowing more noise. Finally, we weaken robustness to f -weak robustness (**WR**), where f is a non-decreasing function: **In the first n selected predictions, no rater may introduce noise larger than $f(n)$** . Weak robustness is a generalisation of strict robustness (let f be a constant function). Here, good (selected) predictions from the past give the rater some credit. Picking $f(n) = n \cdot \theta$ would encode that the average noise is limited by θ . Whether weak robustness is sufficient is debatable, but we should expect increased effectiveness for some f .

We formulate a radically different notion of robustness, based on hypothesis testing. The idea is that one initially assumes perfect accuracy (null hypothesis), and that the null hypothesis may be rejected in favor of *any* alternative hypothesis if the data is unlikely to fit the predictions. The hypothesis testing variant of robustness is (**HR**): **The probability of a sequence of events, given that the predictions are accurate, must not go below α** . We show later that **SR** = **HR**, when $\alpha = -2^\theta$.

Remark 1. Note that we are not subjecting the rater to a single statistical test, but to many. Then, we require that the rater cannot fail any of the statistical tests. This models the notion that we do not know what kind of attack the rater

may be performing (i.e. what the alternative hypothesis is). For every sequence of outcomes there is one statistical test, where H_0 is that the rater accurately predicts it, and H_1 is that the rater underreports it. For each *individual* statistical test, the probability of falsely rejecting H_0 is bounded by α . Since we have multiple tests, the probability that at least one test rejects H_0 can (and does) exceed α .

3.2 Auxiliary Requirements

The system that needs to be robust must also have a variety of other properties. The filter should not introduce bias, it must not rely on foreknowledge, and it must not exclude excessively many predictions.

The first requirement is that the system must be implementable – it cannot make decisions based on future events. Specifically, users cannot be prescient (**T**): **Whether a prediction is used should not depend on its outcome, nor on future predictions or outcomes.** There are combinations of requirements that are logically non-contradicting, but that contradict **T**. Rejecting **T** means asserting prescience. Such systems cannot be implemented in a meaningful way, since the purpose of the prediction was to be able to act before the future happened. Note that **T** does not exclude analysing a prediction in the future, it just prohibits users from using such a future analysis in the present.

Another property of a selection mechanism is that it should not be biased. The ideal notion of unbiasedness, called strictly unbiased (**SU**) states: **A prediction from a user about an event is used iff any alternative prediction from that user about that event would be used too.** However, this notion may be too strong, as the mere possibility of an extreme prediction that may introduce an unacceptable amount of noise would imply that all predictions must be blocked. Hence we formulate (weakly) unbiased selection (**WU**): **A prediction from a user about an event is used iff the opposite prediction from that user about that event would be used too.** This notion matches the idea that we can not “prefer” one outcome over the other, and thus that the selection mechanism mistakenly favours one side. However, weak unbiased selection may introduce a bias towards the center, meaning unlikely events may be overestimated.

Finally, the property that forms the typical trade-off with robustness: effectiveness. Effectiveness measures how many predictions can be used over the lifetime of a trust system. We formulate two incarnations of effectiveness. The first is optimistic k, n -effectiveness (**OE**): **It is possible to select k predictions for n events.** Optimistic k, n -effectiveness can be used to prove hard limits on robustness of trust systems. The second notion of effectiveness is realistic k, n -effectiveness (**RE**): **Assuming all raters are accurate, we can expect k predictions for n events.** The realistic k, n -effectiveness is used for the positive results.

4 Modelling

Raters send predictions to users – be it by broadcasting or upon request. Predictions concern events with an outcome that will eventually be known. Users want to estimate how likely outcomes of events are, and use predictions for this purpose. After the event, users use the outcome to judge the predictors. Good predictors assign high likelihood to actual outcomes, and bad predictors assign lower likelihood.

There is a sequence P of binary events, where the i^{th} event, denoted p_i , either equals 0 or 1. The prediction of rater a about p_i is r_i^a , which (for honest raters) represents his estimate of $p(p_i=1)$ – and $\overline{r_i^a} = p(p_i=0) = 1 - p(p_i=1)$. The sequence of all predictions of rater a is R^a , with R_i^a his prediction about the i^{th} event. For a set of raters A , we can write R^A to mean $\{R^a | a \in A\}$. Together A , P and R^A form a trust system.

The user has no influence on the values of the predictions or on the outcomes of the events. The only way to achieve the goal of dealing with predictions in a robust manner, is to select the right predictions from the predictions that are given. Note that blocking raters can be accomplished by never selecting that rater’s predictions, regardless of the values. Thus, the focus on this paper is on selecting the right predictions. The sequence of predictions that is selected is called the sequence of *filtered* predictions, denoted \widehat{R}^A (where R^A is the set that \widehat{R}^A is selected from).

Our motivating question is what the limitations are to such a filter. The filtered predictions may be biased, can we avoid such a bias? All things considered equal, a looser filter is superior, as it allows the user to consider more information. How many (sufficiently unbiased) predictions can \widehat{R} contain? Finally, the crucial question, can we put a hard limit on how much noise a rater can introduce?

Every prediction has an amount of information [11]. Information is the dual of entropy, and entropy is the expected surprisal of a random variable [5]:

Definition 1. *Let X, Y, Z be discrete random variables.*

The surprisal of an outcome x of X is $-\log(P(x))$.

The entropy of X is $H(X) = \mathbb{E}_X(-\log(P(x))) = \sum_i P(x_i) \cdot -\log(P(x_i))$.

Once the actual outcome p_i of the event is known, we can analyse the surprisal of the prediction, which is $-\log r_i^a$ or $-\log \overline{r_i^a}$, when $p_i = 1$ or $p_i = 0$, respectively. The surprisal of r_i^a given the outcome p_i is denoted $f^i(r_i^a, p_i)$ (to avoid the case distinction for p_i). With perfect information (zero entropy), the surprisal is 0, so surprisal measures noise (misinformation).

Therefore, surprisal can be used to measure the quality of a prediction (this is, e.g., the basis of the cross-entropy method [5]). A high quality prediction assigns a higher probability to the actual outcome. But more importantly, a prediction is of low quality when a low probability is assigned to the outcome. Since a high surprisal corresponds to low quality predictions, we use surprisal to measure the noise of a prediction. However, a high degree of noise in the prediction does not necessarily mean that the rater was malicious or even in error. Raters can introduce noise by sheer bad luck.

Other measures could be used than surprisal. There are, however, two advantages being logarithmic: First, the sum of the surprisal of two outcomes of independent events is equal to the surprisal of the outcome of the joint event. The surprisal of a combination of outcomes is the sum of the surprisal of the individual outcomes; formally $\log p(x) + \log p(y) = \log p(x, y)$, for independent X, Y . Second, it matches the intuition that the difference between 1% and 2% is far more significant than the difference between 50% and 51%. However, we also consider another measure for the quality of predictions, which is based on hypothesis testing; a statistical tool, rather than information theoretic.

Before continuing, we define a couple of shorthand notations. Typically, we denote predictions with r , but we may use q instead. Furthermore, we allow substitution in a sequence/set, denoted $R^A[r_i^a \setminus q_i^a]$, where r_i^a is replaced by q_i^a in R^A . We may want to get the index of r_i^a in the sequence \widehat{R} , which we denote as $\rho_{\widehat{R}}(r_i^a)$. Finally, $X \sqsubseteq Y$ if X is a subsequence of Y (same elements and order). These notations are particularly introduced to simplify the notation of the axioms.

5 Axioms

We need to have a formal model of the trust system to base the formal version of our axioms on. The idea here follows the standard approach in social choice. We formulate a generic collection of events and predictions, and prove that set of filtered predictions can satisfy a certain combination of axioms. Thus, we can show the impossibility of a combination of desirable properties.

Axiom **AR** – absolute robustness – must encode that “no rater may introduce noise.” That axiom can be stated as:

$$\mathbf{AR} : \quad \forall_{i,a} \sum_{r_i^a \in \widehat{R}^a} f^f(r_i^a, p_i) = 0.$$

Axiom **SR** – strict robustness – must encode that “no rater may introduce noise larger than θ .” That axiom can be stated as:

$$\mathbf{SR} : \quad \forall_{i,a} \sum_{r_i^a \in \widehat{R}^a} f^f(r_i^a, p_i) \leq \theta.$$

Axiom **WR** – weak robustness – must encode that “in the first n selected predictions, no rater may introduce noise larger than $f(n)$.” That axiom can be stated as:

$$\mathbf{WR} : \quad \forall_{n,a} \sum_{r_i^a \in \widehat{R}^a \wedge \rho_{\widehat{R}^a}(r_i^a) < n} f^f(r_i^a, p_i) \leq f(n).$$

Axiom **HR** – hypothesis testing-based robustness – must encode that “the probability of a sequence of events, given that the predictions are accurate, must not go below α .” That axiom can be stated as:

$$\mathbf{HR} : \quad \forall_a \left(\prod_{r_i^a \in \widehat{R}^a} r_i^a \geq \alpha \right).$$

The product of the prediction assigned to the actual outcome is what the joint

probability of the outcomes would be if the predictions are accurate. This probability may not go below α .

Axiom **T** – non-prescience – must encode that “whether a prediction is used should not depend on its outcome, nor on future predictions or outcomes.” The axiom can be stated as:

$$\mathbf{T} : \quad \forall_{i \leq k, a} (r_i^a \in R^a = r_i'^a \in R'^a) \wedge \forall_{i < k} (p_i \in P = p_i' \in P') \implies (r_k^a \in \widehat{R}^a \Leftrightarrow r_k^a \in \widehat{Q}^a),$$

whenever two trust systems are equal up to point k , they must allow the same predictions to be selected or blocked. In other words, at time k the selection cannot depend on p_{k+j} or r_{k+j}^a , since there exists a system identical up to k steps with $p_{k+j} \neq p'_{k+j}$ and $r_{k+j}^a \neq r'^a_{k+j}$.

Axiom **SU** – strong unbiasedness – must encode that “a prediction from a user about an event is used iff any alternative prediction from that user about that event would be used too.” The axiom can be stated as:

$$\mathbf{SU} : \quad \forall_{i, a} (r_i^a \in \widehat{R}^a \wedge Q^a = R^a[r_i^a \setminus r_i'^a] \implies r_i'^a \in \widehat{Q}^a),$$

every prediction from user a at time i can be replaced by another prediction from a at i .

Axiom **WU** – weak unbiasedness – must encode that “a prediction from a user about an event is used iff the opposite prediction from that user about that event would be used too.” The axiom can be stated as:

$$\mathbf{WU} : \quad \forall_{i, a} (r_i^a \in \widehat{R}^a \wedge Q^a = R^a[r_i^a \setminus \overline{r_i^a}] \implies \overline{r_i^a} \in \widehat{Q}^a),$$

every prediction from user a at time i can be replaced by another prediction from a at i .

Axiom **OE** – optimistic effectiveness – must encode that “it is possible to select k predictions for n events.” The axiom can be stated as:

$$\mathbf{OE} : \quad \forall_a (\max_{i < n} \rho_{\widehat{R}^a}(r_i^a) \geq k),$$

there highest index of a prediction in \widehat{R}^a with index below n in R^a is at least k .

Axiom **RE** – realistic effectiveness – must encode that “assuming all raters are accurate, we can expect k predictions for n events.” The axiom can be stated as:

$$\mathbf{RE} : \quad \forall_{a, \widetilde{R}^a \sqsubseteq R^a} (\max_{i < n} \rho_{\widehat{\widetilde{R}^a}[\widetilde{R}^a \setminus \overline{\widetilde{R}^a}]}(r_i^a) \geq k),$$

which is similar to **OE**, except it must also hold if we swap arbitrary values of r_i^a for their negation. With the arbitrary swapping of predictions, **RE** captures the possibility that the actual outcome was \overline{p}_i , in which case the surprisal would be $-\log(\overline{r_i^a})$, rather than $-\log(r_i^a)$. Thus, the effectiveness here is attainable for all sequences of outcomes, rather than just one.

6 Relative Strength of the Axioms

With the exception of Theorem 1, all the propositions and corollaries in this section are straightforward sanity proofs. Propositions 1, 2, 3 and 4 and Corollaries 1 and 2 merely show that axioms that are supposed to be weaker are indeed weaker. The relative strength of the axioms is depicted in Fig. 1.

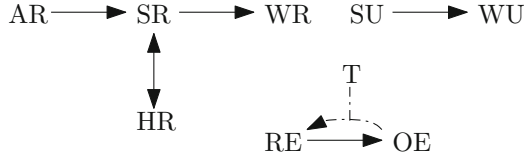


Fig. 1. Relations between axioms. Arrows point from strong to weak.

Theorem 1 is the only deep result in this section, as it shows the equivalence between $\mathbf{SR}(\theta)$ and $\mathbf{HR}(\alpha)$, for $\alpha = \frac{1}{2^\theta}$. Thus Theorem 1 shows that an information-theoretic perspective coincides with a view based in statistical methods; specifically hypothesis testing.

The first proposition shows that a lower fixed robustness threshold is a stronger requirement:

Proposition 1. *If $\theta \leq \theta'$, then $\mathbf{SR}(\theta) \implies \mathbf{SR}(\theta')$.*

Proof. By transitivity: $\forall_{i,a} \sum_{r_i^a \in \widehat{R}^a} f^f(r_i^a, p_i) \leq \theta \leq \theta'$.

Proposition 1 shows that strict robustness is a weaker requirement than absolute robustness:

Corollary 1. *For all θ , $\mathbf{AR} \implies \mathbf{SR}(\theta)$.*

The second proposition shows that a consistently lower robustness threshold is a stronger requirement:

Proposition 2. *If, for all n , $f(n) \leq f'(n)$, then $\mathbf{WR}(f) \implies \mathbf{WR}(f')$.*

Proof. By transitivity: $\forall_{i,a} \sum_{r_i^a \in \widehat{R}^a} f^f(r_i^a, p_i) \leq f(n) \leq f'(n)$.

Proposition 2 shows that weak robustness is a weaker requirement than strict robustness:

Corollary 2. *If $f(1) \geq \theta$, then $\mathbf{SR}(\theta) \implies \mathbf{WR}(f)$.*

The third proposition shows that no bias towards any prediction is a stronger requirement than no bias w.r.t. the opposite prediction:

Proposition 3. $\mathbf{SU} \implies \mathbf{WU}$

Proof. The term \overline{r}_i^a in \mathbf{WU} is an instance of r_i^a in \mathbf{SU} , and \mathbf{SU} dictates that substitution can be done for all r_i^a .

The fourth proposition shows that realistic effectiveness is a stronger requirement than optimistic effectiveness:

Proposition 4. $\mathbf{RE} \implies \mathbf{OE}$

Proof. In \mathbf{RE} , we can find \mathbf{OE} by letting $\widetilde{R}^a = \emptyset$.

Finally, this section's main theorem, which shows the deep link between information-theoretic robustness and hypothesis testing robustness:

Theorem 1. *If $\alpha = \frac{1}{2^\theta}$, then $\mathbf{SR}(\theta) \leftrightarrow \mathbf{HR}(\alpha)$.*

Proof. Note that $\prod_{r_i^a \in \widehat{R}^a} r_i^a \geq \alpha$ iff $\log(\prod_{r_i^a \in \widehat{R}^a} r_i^a) \geq \log(\alpha)$. Distributing the log over the product and negating, $-\sum_{r_i^a \in \widehat{R}^a} \log(r_i^a) = -\log(\alpha)$. This is $\mathbf{SR}(\theta)$ with $\theta = \log(1/\alpha)$.

7 Impossibility Results

Here, we study the relationship between the axioms. Specifically, we investigate whether certain combinations of axioms admit a non-trivial set of filtered ratings. Moreover, where applicable, we investigate what the size of the set of filtered ratings can be. Any statement made in this section is a general truth about all rating systems. The results are summarized in Table 1.

The first is the effective impossibility of a system that has absolute robustness. The only ways in which a system can be absolutely robust, is if it either never uses predictions or if it only uses predictions that predict 100% probability for the correct outcomes. The former implies an effectiveness of 0 (i.e. it is *ineffective*); the latter breaks non-prescience. An absolutely robust trust system without prescience is ineffective:

Theorem 2. $\mathbf{AR} + \mathbf{T} + \mathbf{OE}(k, n) \implies k = 0$

Proof. Let \widehat{R}^a be a subset of R^a such that it satisfies **AR**. Since noise is a positive quantity, $\forall_{r_i^a \in \widehat{R}^a} f^f(r_i^a, p_i) = 0$. Thus $r_i^a = 1$ iff $p_i = 1$. If \widehat{R} is non-empty, then we can take such a, i . Due to **T**, when $p_i = 0$, \widehat{R} remains the same up to i . However, if $p_i = 0$, then $r_i^a = 1$ implies $f^f(r_i^a, p_i) = \infty > 0$. By **T**, if \widehat{R} is non-empty, then there exists a system that violates **AR** or **T**. Hence, $\widehat{R} = \emptyset$ and $k = 0$.

This theorem (and the following) can be stated as an impossibility theorem:

There is no non-prescient, effective, absolutely robust trust system.

Moreover, even if we drop non-prescience (thus using predictions given foreknowledge), the system would not even be weakly unbiased unless all predictions are ignored. In other words, if we select 100% correct predictions, we would lose (weak) unbiasedness. A weakly unbiased absolutely robust trust system is ineffective:

Proposition 5. $\mathbf{AR} + \mathbf{WU} + \mathbf{OE}(k, n) \implies k = 0$

Proof. Similar to to Theorem 2, except rather than swapping p_i , we swap r_i^A .

There is no unbiased, effective, absolutely robust trust system.

Weakening the robustness requirement to strict robustness, we finally obtain a bit of robustness. A non-prescient rating system with strict robustness can allow at most θ ratings to be selected from users:

Theorem 3. $\text{SR}(\theta) + \mathbf{T} + \text{OE}(k, n) \implies k \leq \theta$

Proof. Let \widehat{R}^a be a sequence of r_i^a . Due to axiom **T**, the choice of r_i^a is independent of p_i . Thus, if $r_i^a \neq 1/2$, then the model must hold with noise $f^{\sharp}(r_i^a, p_i)$ and $f^{\sharp}(r_i^a, \overline{p}_i)$. Without loss of generality, we can therefore assume $r_i^a \leq \overline{r}_i^a$. Now, via $\text{SR}(ic)$, $\theta \geq \sum_{r_i^a \in \widehat{R}^a} f^{\sharp}(r_i^a, p_i) \geq \sum_{r_i^a \in \widehat{R}^a} f^{\sharp}(1/2, p_i) = k$

There is no non-prescient, unboundedly effective, strictly robust trust system.

An interesting academic question is whether the fixed bound on effectiveness can be lifted when we are aware of the future. It turns out that if we replace non-prescience with weak unbiasedness, that the bound is widened, but still fixed:

Theorem 4. $\text{SR}(\theta) + \mathbf{WU} + \text{OE}(k, n) \implies k \leq 2^\theta$

Proof. As **T** is not an axiom, we can select r_i^a knowing p_i . However, due to **WU**, $f^{\sharp}(\widehat{R}_{\leq k}^a, p) + r_i^a$ must be at most θ . Let $c_k = \theta - f^{\sharp}(\widehat{R}_{\leq k}^a, p)$. Then we obtain the recursive equation $c_k + \log(1 - \frac{1}{2^{c_k}}) = c_{k-1}$. Via $1 - \frac{1}{2^{c_k}} = 2^{c_k-1-c_k}$, and $2^{c_k} - 1 = 2^{c_{k-1}}$ that becomes $c_k = \log(2^{c_{k-1}} + 1)$. Basic arithmetics show that $c_k = \log(k)$.

There is no unbiased, unboundedly effective, strictly robust trust system.

When tightening the requirement on unbiasedness to strong unbiasedness, we lose effectiveness completely. Even without non-prescience. Thus, strict robustness and strong unbiased cannot be meaningfully combined.

Theorem 5. $\text{SR}(\theta) + \mathbf{SU} + \text{OE}(k, n) \implies k = 0$

Proof. Let r_i^a in \widehat{R}^a , then the theorem must also hold for q_i^a . However, if we let $q_i^a < -\log(\theta)$, and $p_i = 1$, then the strict robustness is broken. Hence, there cannot be any $r_i^a \in \widehat{R}^a$, and $k = 0$.

There is no strongly unbiased, effective, strictly robust trust system.

Again, we weaken the robustness requirement. When we keep strong unbiasedness, we again lose effectiveness. Thus, not a single notion of robustness can combine meaningfully with strong unbiasedness.

Theorem 6. $\text{WR}(\theta) + \mathbf{SU} + \text{OE}(k, n) \implies k = 0$

Proof. Reuse the proof of Theorem 5, replacing θ with $f(1)$.

There is no strongly unbiased, effective, weakly robust trust system.

Finally, we consider a weakly robust, non-prescient system. Here, the limitation on the effectiveness is the weakest (assuming $\theta = f(1)$):

Theorem 7. $\text{WR}(f) + \mathbf{T} + \text{OE}(k, n) \implies k \leq f(n)$

Proof. Reuse the proof of Theorem 3, replacing θ with $f(n)$.

There is no non-prescient, unlimited effective, weakly robust trust system.

8 Robust Prediction Systems

We have shown the negative impact of robustness on other desirable requirements on a rating system. Perhaps robustness is simply a problematic notion in itself. In the proofs, we have shown that models cannot exist in certain combinations, not that models do exist in the negation. In this section, we show that there do exist reasonable models that strike a balance between robustness, fairness and effectiveness.

It does not suffice to prove the converse of the impossibility theorems, as that would simply prove that there exists a set of filtered ratings of a certain size. However, the setting in which that size is reached may be a pathological case. We want to show that filters can be *expected* to achieve a certain size. Hence, we are using axiom **RE**, rather than **OE**. Realistic effectiveness is an assertion about raters whose ratings correspond to the true probabilities. We want to show that these honest raters are expected to have a certain number of ratings selected by the filter.

First we introduce an auxiliary lemma, that shows that under **T**, **RE** = **OE**:

Lemma 1. $\mathbf{T} + \mathbf{RE}(k, n) \Leftrightarrow \mathbf{T} + \mathbf{OE}(k, n)$

Proof. For given \widehat{R}^a , if r_i^a is in \widehat{R}^a , then, via **T**, $\overline{r}_i^a \in \widehat{Q}^a$. We can take \widetilde{R}^a , and swap all r_i^a as above. Then we can make **OE** against any individual instance of **RE** for \widetilde{R}^a . Thus, for **OE**(k, n) and **T** to hold, **RE**(k, n) can be deduced to hold too. Together with Proposition 4, that proves the lemma.

The following theorem concerns strict robustness. A non-prescient, weakly unbiased, strictly robust filter can be expected to have over θ ratings selected over a lifetime:

Theorem 8. *There is a model that satisfies $\mathbf{T} + \mathbf{WU} + \mathbf{SR}(\theta) + \mathbf{RE}(\theta - 1, n)$, for sufficiently large n .*

Proof. Via Lemma 1, it suffices to prove for $\mathbf{T} + \mathbf{WU} + \mathbf{SR}(\theta) + \mathbf{OE}(\theta - 1, n)$. If we only select those ratings that are within distance ϵ from $1/2$, then the noise is at most $k + k \cdot \epsilon$. Letting $k = \theta - 1$, the noise is at most $\theta - 1 + (\theta - 1) \cdot \epsilon$, which is under θ for sufficiently small ϵ . It is straightforward to verify that this scheme does not violate **WU**.

For the next theorem, we look at an interesting subclass of weak robustness. We consider only those functions where $f(i) - f(i - 1)$ is constant; specifically, 1. Thus, every prediction, the rater gets an additional bit credit. If the rater randomly provides ratings, the expected loss equals the gain and a bad rater is expected to make a nett loss. Specifically, the expected change in nett score is $f^{\sharp}(r_i^a, p_1) - 1$, which can be negative, 0 or positive.

Theorem 9. *Raters whose ratings do not correlate with the events, or correlate negatively, have a finite effectiveness. Raters whose ratings correlate positively with the event have a non-zero probability of infinite effectiveness.*

Proof. This is a simple application of a rule in random walks [7]. The probability of ruin – losing all credit – is 1 for random walks with $\mathbb{E}(\text{step}) \leq 0$, and the probability of ruin is strictly below 1 for random walks with $\mathbb{E}(\text{step}) > 1$.

Theorem 9 is a superficially surprising result. We have a hard guarantee that below average predictors are eventually unable to get their ratings selected. We cannot guarantee that a high quality predictor is not shunned too. If a high quality predictor is unlucky, he can still have a random walk ending in ruin. Note that for simplicity, we took the cutoff at $1/2$, we could have chosen arbitrary values, or even a dynamic version. In all these cases, random walks without expected gain eventually run into ruin.

9 Conclusion

We have presented a simple formal model for prediction systems. That formal model focusses on the actual predictions, the outcomes and which predictions are used, and ignores the non-essential aspects of a prediction system.

We have outlined desirable properties for such a prediction system to have. Specifically, we have three notions of robustness – how much noise an attacker (or any rater) can introduce – in various strength (absolute, strict, weak). All three notions are formulated in information theory, but strict robustness can be stated in classical statistical term too. Two notions deal with bias, the strong version disallows any form of bias, whereas the weak version allows bias towards the center. Two more notions deal with effectiveness – how often the user can use the ratings. One version (weak) overestimates the effectiveness, to strengthen the impossibility results. Finally, one notion deals with the fact that users should not be able to foreknow the future.

All these notions have been translated into axioms in the language of the formal model for prediction systems. We show that the axioms do indeed satisfy the desired strength relations.

Based on the axioms, we present a collection of impossibility results. The absolute notion of robustness cannot have any effectiveness whatsoever. The strict notion of robustness can have a bounded effectiveness, meaning that the system cannot keep providing useful predictions indefinitely. For the weak notion of robustness the effectiveness remains hampered.

Finally, we show that a strict robust system can exist, and while its life-span is limited, reaching the theoretically maximal effectiveness is feasible. More importantly, we show that if we weaken robustness, an interesting property regarding effectiveness arises. Selecting the right function ($f(n) = n + \theta$), we get that better-than-random raters could have infinite effectiveness, whereas random-or-worse raters have finite effectiveness. In other words, the quality of the ratings determines whether the effectiveness is bounded.

Together the results in this paper sketch the idea that fairly strong notions of robustness are feasible, but come at a high cost. An interesting research direction would be to fine-tune all the desirable properties into an actual system – rather than a theoretically induced model.

References

1. Arrow, K.J., Forsythe, R., Gorham, M., Hahn, R., Hanson, R., Ledyard, J.O., Levmore, S., Litan, R., Milgrom, P., Nelson, F.D., et al.: The promise of prediction markets. *Science* **320**(5878), 877 (2008). New York, Washington
2. Arrow, K.J.: *Social Choice and Individual Values*. Cowles Foundation Monographs Series. Yale University Press, New Haven (1963)
3. Berg, J.E., Nelson, F.D., Rietz, T.A.: Prediction market accuracy in the long run. *Int. J. Forecast.* **24**(2), 285–300 (2008)
4. Buckley, P., O'Brien, F.: The effect of malicious manipulations on prediction market accuracy. *Inf. Syst. Front.* 1–13 (2015). <http://link.springer.com/article/10.1007/s10796-015-9617-7>
5. Cover, T.M., Thomas, J.A.: Entropy, relative entropy and mutual information. In: *Elements of Information Theory*, pp. 12–49 (1991)
6. Deck, C., Porter, D.: Prediction markets in the laboratory. *J. Econ. Surv.* **27**(3), 589–603 (2013)
7. Feller, W.: *An Introduction to Probability Theory and Its Applications*, vol. I. Wiley, London, New York, Sydney, Toronto (1968)
8. Green, K.C., Armstrong, J.S., Graefe, A.: Methods to elicit forecasts from groups: Delphi and predictionmarkets compared (2007)
9. Hanson, R., Oprea, R., Porter, D.: Information aggregation and manipulation in an experimental market. *J. Econ. Behav. Organ.* **60**(4), 449–459 (2006)
10. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.* **42**(1), 1 (2009)
11. Jianshu, W.E.N.G., Chunyan, M.I.A.O., Angela, G.O.H.: An entropy-based approach to protecting rating systems from unfair testimonies. *IEICE Trans. Inf. Syst.* **89**(9), 2502–2511 (2006)
12. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **43**(2), 618–644 (2007)
13. Kerr, R., Cohen, R.: Smart cheaters do prosper: defeating trust and reputation systems. In: *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 993–1000. International Foundation for Autonomous Agents and Multiagent Systems (2009)
14. Snowberg, E., Wolfers, J., Zitzewitz, E.: Partisan impacts on the economy: evidence from prediction markets and close elections. Technical report, National Bureau of Economic Research (2006)
15. Taylor, A.D.: *Social choice and the mathematics of manipulation*. Cambridge University Press, Cambridge (2005)
16. Wang, D., Muller, T., Irissappane, A.A., Zhang, J., Liu, Y.: Using information theory to improve the robustness of trust systems. In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 791–799 (2015)
17. Wang, D., Muller, T., Zhang, J., Liu, Y.: Is it harmful when advisors only pretend to be honest? In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (2016)

Short Papers

I Don't Trust ICT: Research Challenges in Cyber Security

Félix Gómez Mármol¹, Manuel Gil Pérez^{2(✉)}, and Gregorio Martínez Pérez²

¹ NEC Europe Ltd., Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
felix.gomez-marmol@neclab.eu

² Departamento de Ingeniería de la Información y las Comunicaciones,
University of Murcia, 30071 Murcia, Spain
{mgilperez,gregorio}@um.es

Abstract. Can we trust ICT (Information & Communication Technology) systems? Every single day a handful of previously unknown security vulnerabilities on these environments are published, dangerously feeding the lack of trust feeling that many end users already exhibit with respect to ICT. In order to disrupt and even invert such a perilous tendency (hindering the wide adoption of ICT and all its associated benefits), a number of research challenges in the field of cyber security need to be addressed. This paper presents some of these key challenges, offering initial thoughts on how to tackle each of them.

Keywords: Trustworthy ICT · Cyber security · Research challenges

1 Introduction

The numerous benefits brought by Information and Communications Technologies (ICT) are unquestionable today. Yet, a non-negligible amount of end users feel often reluctant to enjoy those advantages, since they distrust such ICT. And this lack of confidence is mainly due to the perception of insecurity that the ICT systems pose. Despite the large amount of works and efforts mainly done by the research community, government agencies and industry, oriented to provide security solutions for the ICT systems [1], everyday we observe fateful news regarding the proliferation of new cyber attacks, thefts, threats and other potential cyber crimes. Thus, we state that such mistrust will persist while the aforementioned perception of insecurity in ICT systems remains [2].

In this context, this paper presents some of the main challenges in the cyber security field that must be first addressed and solved in order to increase the trustworthiness of the end users in the ICT systems, in a way that the former may benefit from the latter. It is noteworthy that this paper does not merely list a number of challenges, but it also provides a pool of initial ideas on how to manage each of them. Therefore, the main contribution of the paper is to bring together a number of challenges in order to foster and encourage research in the

field of cyber security, with the ultimate goal of increasing the trustworthiness deposited by end users in ICT systems.

As stated before, cyber security entails a large list of challenges, where the most critical ones can be grouped in the following four main research trends. They have to be treated adequately in order to provide greater trustworthiness of the end users when using the ICT systems.

- **Dynamic risk management.** The organizations' operational needs have to be continuously tackled to update the risk level of any change happening in the corporation on its assets: changes in threats, new vulnerabilities, new response actions or countermeasures, or even modification of the assets themselves [3]. A dynamic risk management or treatment system requires a continuous feedback mechanism to monitor threats in a real-time basis, and so allowing a quick reaction to minimize the exposure time in front of potential risky situations and events for the organization being protected.
- **Attack and defense graphs.** One of the main ways of providing risk assessment is supporting the implementation of attack and defense graphs [4]. With them, the dynamic risk management systems pretend to estimate the level of risk of the assets through the definition of attack patterns to capture dynamics of a threat and stages it has to go through.
- **Incidents correlation.** The correlation mechanisms are a required feature to reach a holistic view about the cyber security of any organization. All the sensors, strategically deployed in the underlying network, should share their monitoring information in an orchestrated way with the aim of correlating the individual evidences detected by each of them in different locations [5]. With this information, the dynamic risk assessment engines will subsequently compute the instantaneous risk level of the organization at any time.
- **Information sharing.** In the current distributed systems, it is necessary to define an information model with which to exchange the corresponding information between the different stakeholders in order to detect distributed threats [6]. This will require the design and deployment of context-aware security and privacy models protecting the process of sharing information among the different actors of the dynamic risk management system: how to securely share the information, which one can be shared and which one cannot. Furthermore, the risk information sharing conveys the use of standard formats and protocols to reach a common assurance model between stakeholders in a trustworthy way.

All these challenges can be summarized as follows, where the text in bold corresponds to the previous main four research trends and the italic text represents the properties of each of them:

Dynamic risk management over large systems using *adaptive attack/defense graphs* with *privacy-preserving incidents correlation* and *encouraging information sharing*

In the next sections, we present the main research challenges in the cyber security field regarding the four research trends enumerated earlier.

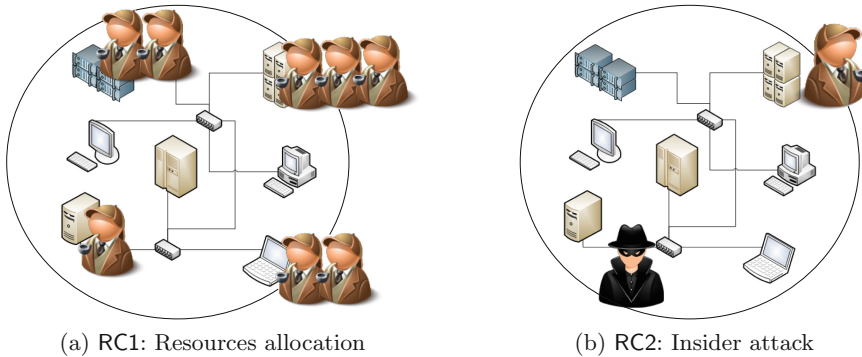


Fig. 1. Resources allocation and insider attack cyber attacks

2 Dynamic Risk Management

Despite the plethora of works aimed at designing accurate dynamic risk management in ICT systems, there are still unresolved research challenges (**RC**) that should not be neglected. Among them, we highlight the following main challenges that, in our opinion, represent initial ideas to be firstly addressed them.

RC1. *How to estimate how much effort (resources) to put on monitoring/protecting each asset?* In an ICT system the assets are limited, but so are the resources to protect them too. Hence, there is a need to smartly allocate resources to protect each asset (see Fig. 1a). Moreover, such assignment can be dynamic throughout time. To this end, dynamic risk management can become a powerful tool to influence such resources allocation decision.

RC2. *How to minimize the impact of unexpected advantages in a cyber attack (e.g., an insider attack)?* One of the most potentially harmful attacks is the one coming from an insider within the system to be protected (see Fig. 1b). In those cases, where a trust relationship between the insider and the organization is violated, it is critical to minimize the damage inflicted by the attacker. Thus, an appropriate risk management could promptly raise a flag when a suspicious behavior is detected from a user within the system.

RC3. *How to detect cyber attacks trying to divert the victim's attention to protect non-critical assets, while actually compromising critical ones?* Attackers might pretend to be interested in a given asset, trying to force the system to allocate more resources to protect it, while their true interest lies in another asset (see Fig. 2a). These so called reverse honeypots can be effectively combated with an accurate and dynamic risk management, indicating at each time which are the assets under real attack and which not.

RC4. *How to detect back doors inadvertently installed on the system?* Another advanced type of attack consists of surreptitiously installing a so called back door (see Fig. 2b). This intends to be undetectable by the victim, which will be subsequently used to perform an actual attack on the system.

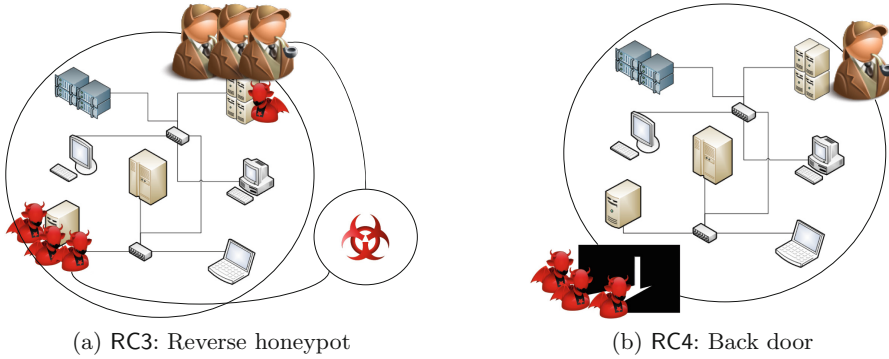


Fig. 2. Reverse honeypot and back door cyber attacks

A comprehensive penetration test can help out to assess the current risk of each asset in the system and, consequently, unveil hidden back doors.

RC5. How to predict a potential cyber attack over a given asset? Ideally, every system administrator would like to predict an attack before it actually happens (see Fig. 3a). In this case, a smart combination of dynamic risk management, attack graphs, incidents correlation and information sharing can be extremely effective to make accurate guesses about imminent attacks.

RC6. How to protect assets against zero-day exploits, while preserving usability/availability? Similarly to RC2, zero-day exploits, by definition, cannot be avoided (see Fig. 3b). Nevertheless, we can (and must) minimize the potential impact that such attacks might have on the system. And here the real challenge is to do so while preserving usability/availability of the protected assets. Again, dynamic risk management can be extraordinarily helpful to achieve such balance between assets protection and availability.

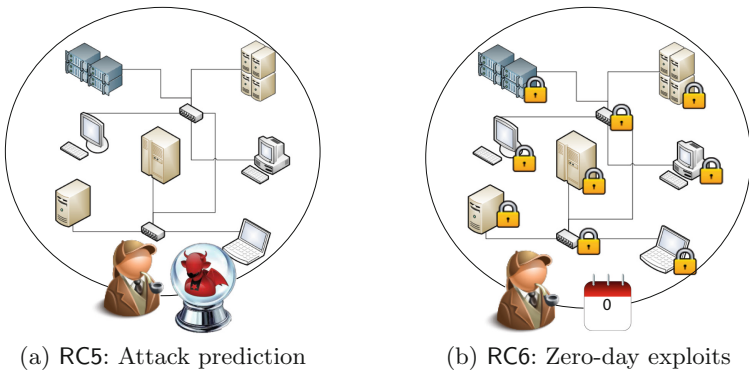


Fig. 3. Attack prediction and zero-day exploits cyber attacks

3 Attack and Defense Graphs

Both attack and defense graphs have captured the attention of many researchers and security experts worldwide. Yet, their notable complexity and modest scalability are still refraining their wide acceptance and deployment [4]. Next we introduce some research challenges regarding attack and defense graphs.

RC7. *How to effect a tailored and adaptive response to a cyber weapon?*

Whereas sophisticated attacks are often tailored to the system they are targeting, the countermeasures applied to defeat them are usually rather generic. To maximize the effectiveness of the response given to an attack, the remedies should be tailored to the specific threat they are facing. To this end, defense graphs, for instance, can constitute an essential aiding tool.

RC8. *How to detect the target of a cyber weapon?* Some generic cyber weapons do not have a specific target and act on an indiscriminate fashion over ICT systems. Yet, some others actually focus on particular environments such as critical infrastructures or enterprises, for example. Promptly identifying the specific ecosystem targeted by an attacker is extremely helpful in deciding how to counter such threat. Thus, attack graphs are capable of indicating which is the most plausible victim of a given cyber weapon.

RC9. *How to detect a dormant/latent cyber weapon in our domain?*

Related to RC4, a cyber weapon can remain on a dormant state, waiting for remote instructions to wake up and perform the actual attack. While in this latent state, it will try to go unnoticed to the administrator of the victim's system. In this regard, attack graphs can assist those administrators to identify both a cyber weapon getting into the dormant state, as well as a latent one receiving commands to wake up imminently.

RC10. *How to detect the self-destruct capability in a cyber weapon and how to prevent it?* Some advanced cyber weapons are equipped with a self-destruction capability, leaving no trace when they realize they have been detected by the target system. In those cases, it is many times a cumbersome task to try to get information about the source of the attack or to learn how to fight against such threat. Again, attack graphs can effectively help to identify the initiation of this self-destruction procedure and abort it.

RC11. *How to avoid an attacker to snoop into a victim's domain in preparation for a cyber attack?* One of the first things an attacker does is to carefully study the victim's domain, seeking for vulnerabilities or weaknesses. Being able to detect such pre-analysis enough in advance gives a very valuable advantage to the administrators when defending against the actual attack. Attack graphs, together with dynamic risk management and incidents correlation, can help in unveiling suspicious behaviors considered as actions conducted by potential attackers in preparation for a cyber attack.

4 Incidents Correlation

Sophisticated attackers no longer play alone. An advanced attack usually consists of multiple steps, either subsequently or concurrently executed that, isolated,

might not be detected as a harmful action, but when combined, they deploy all their damage on the target system. In this regard, incidents correlation can constitute a very effective tool to accurately spot these situations [5].

RC12. *How to detect cyber weapons capable of smartly colluding with other cyber weapons?* A sophisticated cyber weapon might be able to detect other cyber weapons in the victim’s domain and, even more, collude with them to provoke a bigger harm. Here, an appropriate incidents correlation could reveal such perilous collusion with potential devastating consequences.

RC13. *How to discern whether a cyber attacker is controlling certain infrastructure (devices, networks,...) to perform the attack?* Some attackers do not hit their final target directly, but they rather first compromise other systems (known as *botnet*) and then use those to perform the actual attack over the real victim’s domain. Such strategy hinders the identification of the real source of the attack. Yet, a smart combination of incidents correlation and information sharing can ease such identification.

RC14. *How to detect a “composite” cyber weapon smartly split into (apparently) innocuous parts?* Another sign of sophistication in a cyber weapon consists of partitioning it into several (apparently) innocuous pieces. Each of these parts, isolated, is usually harmless (and therefore undetectable by defensive mechanisms), but when combined all together the real damage is inflicted. Again, an intelligent combination of attack graphs and incidents correlation can help in diminishing this specific threat.

RC15. *How to discern whether a cyber weapon is autonomous or being remotely controlled by an attacker?* While certain cyber weapons only react upon a given command from the attacker, others are rather autonomous in their malicious behavior. Being able to detect the first case can help to neutralize the attack by blocking command and control channels used by the cyber weapon to receive its orders. And to achieve that, again a coherent mix of attack graphs and incidents correlation can be of extreme utility.

RC16. *How to detect a multi-vector cyber weapon?* Complex and advanced cyber weapons might not take advantage of just one single approach with the aim of assaulting the victim’s domain, but rather try various entry points. That weapon is known as a multi-vector one, and an appropriate incidents correlation can be crucial to unmask this type of attackers.

5 Information Sharing

It is unrealistic to think of securing ICT systems without sharing relevant data devoted to their protection. Many entities feel reluctant to share certain information that they might consider sensitive arguing privacy concerns [6], as well as other issues like current regulations, as thoroughly discussed in [7].

RC17. *How to detect whether a cyber attack is becoming epidemic?*

Often, when a system is under attack, its administrators are unable to see

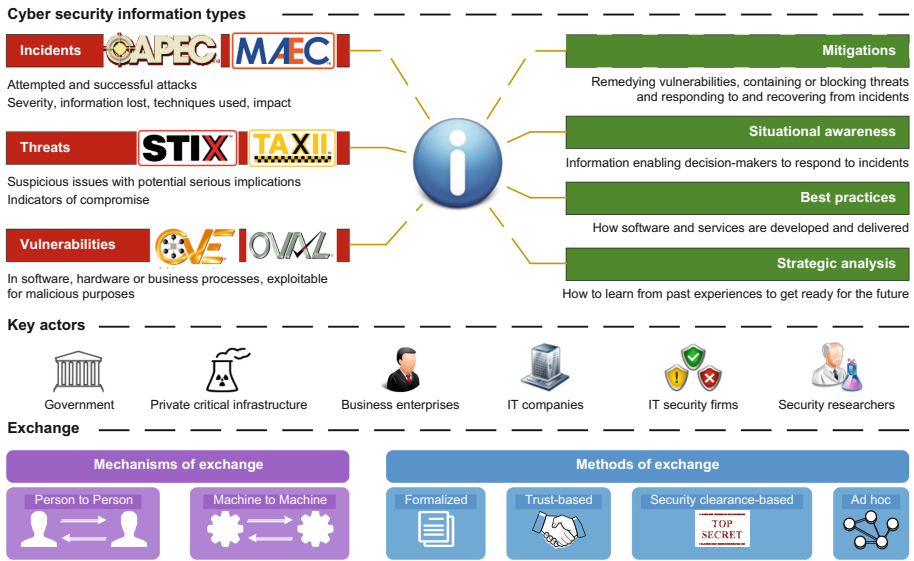


Fig. 4. A framework for cyber security information sharing and risk reduction

beyond the borders of their domain, having thus a constrained view of the overall spreading of a particular cyber weapon. By sharing specific information amongst different realms, it is possible to detect whether a cyber attack is becoming epidemic and, if so, prioritize on battling it back (see RC1).

RC18. *How to effectively and promptly (maybe also automatically and in a standardized way) communicate and share the remedy to a given cyber attack amongst allies?* Nowadays we still face (too often) systems exposed to rather old vulnerabilities for which there is even a patch (maybe also old). The challenge, therefore, is on how to disseminate or propagate these mitigations, patches or bug fixes so that they reach every vulnerable system in a timely fashion.

RC19. *How to incentive information sharing to boost collaborative intrusion detection?* One of the main impediments to a successful information sharing is precisely the reluctance of the participating entities to distribute certain information that, in many cases, might be considered as sensitive or confidential. Hence, an appropriate incentives mechanism should foster such collaboration in detecting cyber attacks.

RC20. *How to identify, with a certain level of confidence, the attacker in a cyber attack?* The so called attribution problem, or how to reliably identify the source of a cyber attack, might be in many cases quite a tough task for administrators. To aim them in overcoming this difficulty, a consistent information sharing strategy, together with the most advanced incidents correlation mechanisms could be enforced.

RC21. *How to measure whether a given domain is susceptible of having cyber weapons or being producing cyber weapons?* Similarly to RC11, if the administrators of a system get to know well in advance that another domain contains or is producing cyber weapons targeting such given system, they can better get ready to counter back such threat.

A number of secure measures, standard formats and potential actors are depicted in Fig. 4 for sharing information, as well as how to lessen the exposure risk by making use of the research challenges identified earlier.

6 Conclusions

This paper has proposed an initial number of research challenges that need to be tackled in order to increase trustworthiness of the end users with respect to the ICT systems. All these challenges deal with cyber security threats that appear everyday incessantly, which we grouped and analyzed into four main research trends, namely: dynamic risk management, attack and defense graphs, incidents correlation and information sharing. With the aim of increasing the trustworthiness of these end users in the ICT systems, we have also provided some initial thoughts on how to deal with each of the aforementioned challenges.

Acknowledgment. This work has been partially supported by the European Commission under grant agreements FP7-ICT-2013.1.4/609062 - SMARTIE (*Secure and Smarter Cities Data Management*) and H2020-ICT-2014-2/671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*).

References

1. Gil Pérez, M., Gómez Mármol, F., Martínez Pérez, G., Gómez Skarmeta, A.F.: RepCIDN: a reputation-based collaborative intrusion detection network to lessen the impact of malicious alarms. *J. Netw. Syst. Manage.* **21**(1), 128–167 (2013)
2. Robinson, M., Jones, K., Janicke, H.: Cyber warfare: issues and challenges. *Comput. Secur.* **49**, 70–94 (2015)
3. Poolsappasit, N., Dewri, R., Ray, I.: Dynamic security risk management using bayesian attack graphs. *IEEE Trans. Dependable Secure Comput.* **9**(1), 61–74 (2012)
4. Abraham, S., Nair, S.: A predictive framework for cyber security analytics using attack graphs. *Int. J. Comput. Netw. Commun.* **7**(1), 1–17 (2015)
5. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. *J. Big Data* **2**(1), 1–41 (2015)
6. Goodwin, C., Nicholas, J.P.: A framework for cybersecurity information sharing and risk reduction. Technical report, Microsoft Corporation, January 2015
7. Skopik, F., Settanni, G., Fiedler, R.: A problem shared is a problem halved: a survey on the dimensions of collective cyber defense through security information sharing. *Comput. Secur.* **60**, 154–176 (2016). <http://dx.doi.org/10.1016/j.cose.2016.04.003>

The Wisdom of Being Wise: A Brief Introduction to Computational Wisdom

Stephen Marsh¹(✉), Mark Dibben², and Natasha Dwyer³

¹ University of Ontario Institute of Technology, Oshawa, Canada
stephen.marsh@uoit.ca

² University of Tasmania, Hobart, Australia
Mark.Dibben@utas.edu.au

³ University of Victoria, Melbourne, Australia
Natasha.Dwyer@vu.edu.au

Abstract. This paper explores how the very human notion of Wisdom can be incorporated in the different behaviour and ultimately reasonings of our computational systems. In particular, it extends and combines previous work in the areas of Computational Trust, Socially Adept Technologies, Device Comfort and the more recent notion of *Slow Computing* that was teased out at a recent Dagstuhl seminar. A brief exposition of Wisdom, its place in autonomous sociotechnical systems, and pointers to how we can make it work are provided. Further work is explored.

1 On Being Wise, and What It *Might* Mean for Computers

the only real wisdom is knowing you know nothing.

Socrates¹

As humans, we value wisdom. It provides, in those who possess it, a knowledge of *how things should be done*, how life should be lived – either to the full or in some way that has less impact on, or is more in touch with, the world and society around it. Its gift is the ability to adapt to new, unforeseen, unexpected happenings gracefully, putting into practice experience in order to manage that which has not been encountered before.

Computers and computational systems do not (presently) possess wisdom. It is an unerringly ‘natural’ phenomenon.² However, we believe that there may be something to be gained from its study, and ultimately the ability to incorporate the behaviours and reasoning into (at least semi-) autonomous computational systems can bring benefit.

¹ A possibly equally old saying has it that Wisdom is knowledge that you gain immediately after you need it. . . .

² While there may be some debate, for which this is almost certainly not the right venue, there appears to be evidence that suggests that other animals than humans may possess it [9].

This short paper *begins* the exploration of wisdom and suggests ways in which Computational Wisdom might be achieved. It begins with an exploration of wisdom in the natural world (which includes people), and searches for common traits. It then delves into how these traits can be identified and aimed for in our computational systems, before beginning an exploration of a possible framework for what we have come to call *Computational Wisdom*, and how it relates to previous work. Finally, and perhaps most importantly, it lays out a set of future goals that we can work towards in the search for truly wise, and as a result resilient and people-focused, computational systems.

As a brief aside, this paper may appear highly conjectural, at least at the outset. We make no apologies for this: it's an unusual topic to think of in computational terms, and the very few examples that have done so (see for example [22]) appear to have done so more from the point of view of understanding the human mind than the uses of the concept. Many more useful expositions can be found in the management sciences, for wisdom is very much a tool and a trait of success [18, 19]. To that end, just as in all our previous work, this is a multidisciplinary journey.

2 On What Has Come Before

This is not the first time 'Wisdom' has been evoked in the computational sphere, and it's certainly not the first time wisdom has been studied in the human, psychologically, philosophically, and in religious studies, to name the three more relevant areas of research and discourse. In this section, we discuss the differences between the former and our positions, whilst the next section discusses the latter.

There are probably three main areas where wisdom has been discussed in the computational. The first is in *Computational Epistemology* [22], the study of knowledge as it relates to and is used by computational systems – a root, in one direction, and a branch, looked at in another, of cognitive science. The second is in Harel et al's proposal for *Wise Computing* [6], the development of systems where the artificial system is capable of being a 'tier 1' member of the designer's club of systems. The third, related and most probably most developed, is the Artificial Sapience [17] movement that sparked in the middle of the first decade of this century, and was focused on knowledge intensive multi-agent systems. Each of these thrusts has something to teach us, but each is ultimately a thrust that attempts to isolate 'wisdom' for explicit use in artificial systems. Our own work acknowledges the place of this, but specifically requires that Wisdom in the Computational sense is a system of systems involving both artificial and 'natural' (or human) – the trick, then, is to know where one wisdom begins and another ends (since wisdom appears to be contextual too [4]).

As well, computing culture has explored one notion of wisdom in the form of the 'wisdom of the crowds' (WOC) movement. However, the type of wisdom espoused by WOC is not what we are referring to in this paper and we wish to differentiate our conceptualisation from common notions of the WOC. Our understanding of wisdom involves a subtle handling of context. In contrast, WOC

refers to a practice of gathering the input from a large amount of users who can bring a wide range of expertise, experience and aggregation methods to a scenario [8]. Applications of the WOC include using a wide range of inputs to filter internet content and improve the results of a search engine [20]. While we acknowledge that the WOC method can harness a large number of perspectives, it is more a form of collective intelligence does not offer the experience we refer to as ‘wisdom’. From a cynical perspective, WOC is simply a way to solve an issue cheaply [24]. Wisdom precedes problem solving and is a form of knowledge that can help reconceptualise a problem situation, including processes such as problem finding and problem making.

Our work differs in some fundamental ways from the previous examinations of wisdom in the artificial. It is not a study of computational epistemology [22], which seeks to explore the means by which we may build truly wise computers, more it is the application of principles of wisdom to the systems that may exist. The difference is, in the spirit of Mark Prensky (see for instance [21]), that true wisdom comes from acknowledging that the answers are not always known, and that the synthesis of ideas (such as computational epistemology seeks in its fulfillment [22]) requires many different inputs, and finally that some of these inputs at the very least are human. Computational Wisdom is a system of systems that is able to understand the value of each part of the system and harness each to their full potential.

On a related note, it’s about, but more than, sapience in the concept of Artificial Sapience. Whilst wisdom and sapience are seen as synonymous, particularly in the Western sphere, we see them as different. Indeed, as [26] notes, Western concepts of wisdom are inherently cognitive, whereas the Eastern concept is *both* cognitive *and* affective. In our other work to date, and in our concept of Computational Wisdom, we acknowledge the affective as a very important aspect of the whole. Knowledge and cognition (as in the Berlin Wisdom framework [1], for instance) are vital, as is metaknowledge, but the *relationship* between observer, actor, ‘user’ (in the sense of a person using’ a computer), system and environment are the key, as can be seen in our Device Comfort work, for instance [15].

Wisdom has also been applied in educational settings, and in Suarez’s thesis [25] we find the concept of Wisdom “by design”, which discusses the notion of artificial wisdom briefly by examining how to design complex social systems that embody wisdom. The work is interesting in that it espouses several principles of wisdom-based design, but its focus on complex societal structures (such as the educational system) means that it is somewhat removed from a computational wisdom as a goal.

Although not a subject for mainstream social science research until the last few decades, the search for an understanding of wisdom has nevertheless seen some important advances. The next section provides an overview of some of the main aspects if this work in the social.

3 On What It *Does* Mean for People

Wisdom is not an unstudied concept, particularly in philosophy, psychology and religion. It is in particular to these disciplines we must turn to more fully

understand the phenomenon. Additionally, as we noted above, the cultural components of wisdom are some things we cannot ignore in our examinations [26, 27].

Wisdom is sought by individuals as means to cope [3]. Similar to notions of trust, where an individual seeks to place their faith in someone or something, wisdom works as a guiding light. Wisdom is not either completely rational or irrational and is a phenomenon that is difficult to explain [3]. It is a ‘big picture’ viewpoint that takes into account a broad perspective that values longer timeframes. As Hoefstede argues [7], Western cultures tend not to be orientated towards longer-term conceptualisations of a situation. The emphasis is more on short-term returns. This may explain why notions of wisdom tend to be associated with cultures outside of the West, such as Asian cultures that are sometimes exoticised by the West.

4 Principles of Wisdom and Their Computational Reflections

We can begin to see a structure here. Wisdom has certain principles (or behavioural and reasoning patterns) associated with it. As in previous work [11, 13, 15] we believe that it is possible to isolate these principles and, furthermore, use them in computational settings. In this section of the paper, we begin the former. The following section begins the rather more detailed work required for the latter.

We have been asked, why these particular principles and not others? The simple answer is that these are the beginnings of what we think are relevant. A more complex answer is: we chose these because they provide building blocks, and sensible starting points for researchers considering Computational Wisdom *per se*. That they are arbitrary is not in question, but we hope they provide food for thought nevertheless.

4.1 Principles

Wisdom Works in the User’s Interests. As we have already noted in [11, 12], computing is about and for people. Part of this ‘equation’ comes with the understanding that, if there is a problem, the solution is not to make life more difficult for people.

Computational Wisdom Creates Calmness for the User. Rather than subject to corporate interests and ‘rational’ computations that ignore the richness of human life, the system is working in the user’s interests. Nuanced contexts are embraced and users are guided through the messy and contradictory demands of everyday life, keeping a focus on the values that really matter.

Wisdom is Slow. The creation of ever more complex autonomous systems can be seen as, and indeed in many instances is, beneficial. However, as we have argued before (see for instance [11]), there are instances where this is not the case. As we note in [10], slowing the system down to a human level when an ‘edge’ case

arises, where systems do not know the ‘answer’, can bring benefits, from greater human understanding to more correct decisions (not to mention reduced liability and an ethical stance that is defensible in real ways). It has been pointed out that slowness need not indeed be a human-centric phenomenon for wisdom. We concur: a slowing down of processing or relaxing of time constraints, to allow for more consideration or simply reflection on the part of a wise system is, we believe, a vital aspect of how future systems, in the right contexts, can and should behave. Since we also believe that all systems ‘touch’ people at some point, this can only benefit the human both ‘in’ and ‘out’ of the loop.

Computational Wisdom is Multi-faceted. As [25] notes, there are different aspects of wisdom that can be designed for, including time-sensitivity, balance (between ‘vision’ and ‘action’) and practicality. From our point of view, this relates to the idea of human-centric systems, where systems should ‘think’ before acting. But the idea of a multi-faceted approach is worth exploring: in this short paper we will not be able to bring all of the facets of wisdom to the fore, and make this a priority for future work.

Wisdom is Adaptive. We have noted above the contextual nature of wisdom. It is true that different circumstances require different actions, but further, Computational Wisdom requires that *ostensibly* the same circumstance may require different actions (for instance, when something new has been learnt by the system as a result of prior actions).

5 How Can We Engage Wisdom: Thoughts on Computational Wisdom

Computational Wisdom is seen here as a system of systems, where the technology *augments* the human in one direction, whilst being *augmented by the human* in another, and finally where different technological systems, such as the computational trust [13, 14, 16] comfort [5, 15] slow computing [10] and socially adept or intelligence technological systems [2] we and others have already postulated and built, build what wisdom is possible between them.

There are, we conjecture, two main questions to answer on the journey to this system of wisdom. The first is *when* it should engage or be engaged, whilst the second is *what form* of engagement is to be exhibited.³

Some of the ideas from the social sciences, concerning what can be said to make up a wise person are difficult to make manifest in the AI realm [18, 19]. These might include: notions of courage and bravery or even perhaps the ability to engage in open productive dialogue; the capacity to appropriately criticise and recognise the need for professional detachment; as well as the capacity to contain one’s emotions in the face of negative feedback or behaviour; emotional empathy; and a well-rounded curiosity for life in general. However, it seems to us that a

³ Of course, the answers to these questions contain an element of wisdom in and of themselves. . . .

whole host of others might be useful in integrating complex computational systems with the capacity to act wisely. For example, it is already well understood that trust can be instigated in such systems for the purpose of building agency. In addition, the very nature of the highly complex systems we are dealing with require the ability for AI agents to make clear decisions (qua judgements) in the face of paradoxes and ambiguity; the formalism for this are an inherent part of information systems.

It seems to us, then, that wisdom within computing insofar as the capacity to deal with uncertainty through statistical solutions to unknowns brings us close to something of what we mean by wisdom within computing systems, much as Sevilla notes [23]. What more is needed? First, some aspect of what we might anthropocentrically call ‘humility’, but in this setting can be seen as computational systems that are capable of taking their own limitations into account, and also of referring to other parts of the system for information when necessary. In addition, since complex systems are already suffused with algorithms for making predictive decisions of desired content, based on learnt knowledge of (e.g.) a user’s browsing behaviour, so it is not a large step to refine these types of algorithms to allow agents to recognise when it is not appropriate to share knowledge with other agents.

Another aspect that McKenna and Rooney suggest as being fundamental to wise leaders, introspection, can also be seen in complex computational systems. That is, they can be configured to recognise errors, seek out reasons for them and use that reasoning to inform future decisions. This, in turn, can be extended to the capacity question established norms, systems, processes and procedures. In addition, the capacity for machine learning provides the means for AI agents to behave as if they were prudent. That is, act in the right way, based on formal logic and reasoning, modified perhaps by algorithms intended to convey human values such as trustworthiness and an understanding of importance and risk.

One aspect of wisdom that may be more difficult to apply concerns the way in which wise judgment takes into account intuition, political acumen, capacity for tact, subtlety and shrewdness, and also may be based on recognition of a higher purpose or the common good. That said, were these pre-determined as part of the complex system’s purpose and the necessary algorithms constructed to focus on that higher end, even this aspect of wisdom may be incorporated. The learnt component of decision making may also enable the system to accommodate wrong answers (i.e. setbacks or disappointments) and behaving as if they were being more cautious in situations entailing risk.

6 The Future, Perhaps

There is, we believe, promise in the short study of wisdom that we have heretofore made. It is a powerful notion with acknowledged strengths. Moreover, it lends itself to a set of principles that we can isolate and begin to work with. Finally, it has within it enough hooks and identifiable requirements to be able to construct a framework around it that allows computational systems to recognise the need

for it, engage its principles, and formulate behaviours that exhibit some at least of its strengths. In short, it is an ideal object of study.

A short paper such as this is a poor exposition of a powerful notion, but its aim is to provide both an inkling of the potential as well as ideas as to how this can be studied and worked towards. We hope that the previous sections have gone some way toward the former. Our future work will take this further, and provide insights into the latter.

6.1 On Future Work

There have been some attempts to define a research agenda to explore wisdom in the computational, most promisingly that of [23] and the notion of Artificial Sapience [17]. In most cases, the goal is an intelligent system that behaves wisely, for some definition of wise. As we have already noted above, we see the limitations here, that such systems are not oriented toward to very people that would work alongside them, and this is a failing.

To address this, we explicitly acknowledge the human in the system as an equal (if not, in this setting, superior) partner. The human is where the system can learn from, and fall back to, in difficult situations, for example. And so, in any study of computational Wisdom, we must start with the human element. Our work is currently taking us in the direction of user engagement with complex sociotechnical systems, automation and information, and this is no exception. Our next step for Computational Wisdom is indeed to set goals in collaboration with human users to ascertain the best uses and directions for ‘wise’ systems to take, what they might look like, and how they might express themselves to their human partners.

6.2 A Conclusion

Notwithstanding certain emotional attributes of wisdom may be perhaps beyond the capacity of complex computational systems, many aspects of wisdom can be modeled and integrated into their decision making processes. Our thesis is this: Wisdom is a difficult multi-faceted construct to grasp (and indeed, to attain, hence its value!). It’s certainly worth studying computationally, and importantly, as we have previously discovered with trust, it might indeed be possible to create computational systems that act *as if* they are wise. This would be a rich extension to systems modeled to act as if they were trusting and trustworthy, because it would bring a wider range of factors to bear upon the system and its outputs. In order to get there, we need to comprehensively study and include the many different spheres of study that have touched on the concept.

Wisdom is powerful, well respected, and in all systems, a goal: we strive to be wise, to behave wisely, and to be seen as wise. This paper argues that the goal of wisdom is something that we can also work towards for the artificial systems that we create to take their place in our societies. Such systems can be more focused on the ‘wise’ thing to do in any given circumstance, and would naturally include the traits of critical thought, the identification of when to engage wisdom and

why (or why not), a focus on the well-being of the society and the people around them, as well as the other systems that exist, and, we conjecture, a calmer and slower approach to reasoning that engages rather than alienates the very people that the systems serve.

It is a long road. This is one of the first steps.

Acknowledgments. We thank the anonymous reviewers for their erudite comments and thoughts, which have helped make our thoughts more concrete. The first author gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada under the Discovery Grants Program.

References

1. Baltes, P., Staudinger, U.: Wisdom: A metaheuristic (pragmatic) to orchestrate mind and virtue toward excellence. *Am. Psychol.* **55**(1), 122–136 (2000)
2. Dautenhahn, K., Bond, A., Canamero, L., Edmonds, B. (eds.): *Socially Intelligent Agents: Creating Relationships with Computers and Robots*. Kluwer Academic Publishers, Dordrecht (2002)
3. Gehrman, K.: Absorbed coping and practical wisdom. *J. Value Inq.* 1–20 (2015)
4. Grimm, S.R.: Wisdom. *Australas. J. Philos.* **93**(1), 139–154 (2015)
5. Guo, J., Jensen, C.D., Ma, J.: Continuous context-aware device comfort evaluation method. In: Damsgaard Jensen, C., Marsh, S., Dimitrakos, T., Murayama, Y. (eds.) *IFIPTM 2015. IFIP AICT*, vol. 454, pp. 203–211. Springer, Heidelberg (2015)
6. Harel, D., Katz, G., Marelly, R., Marron, A.: Wise computing: Towards endowing system development with true wisdom (2015). arXiv preprint <http://arxiv.org/abs/1501.05924>
7. Hoefstede, G.: *Cultures and organizations: Software of the mind*. McGraw-Hill, New York (1997)
8. Hosseini, M., Moore, J., Almaliki, M., Shahri, A., Phalp, K., Ali, R.: Wisdom of the crowd within enterprises: Practices and challenges. *Comput. Netw.* **90**, 121–132 (2015)
9. Lieff, J.: Wise animals: Animal studies need to be in natural settings, not lab (2012). <http://jonlieffmd.com/blog/wise-animals-2>, viewed 20th February 2016
10. Marsh, S.: All human values are system values. Abstract and Talk given at Dagstuhl Seminar on Social Aspects of Self-Organizing Systems, 23 November 2015
11. Marsh, S., Basu, A., Dwyer, N.: Rendering unto cæsar the things that are cæsar’s: Complex trust models and human understanding. In: Dimitrakos, T., McKnight, D.H., Moona, R., Patel, D. (eds.) *IFIPTM 2012. IFIP AICT*, vol. 374, pp. 191–200. Springer, Heidelberg (2012)
12. Marsh, S., Dwyer, N., Basu, A., El-Khatib, K., Storer, T., Esfandiari, B., Renaud, K., Bicacki, M.: Foreground trust as a security paradigm: Turning users into strong links. In: Kayem, A., Meinel, C. (eds.) *Information Security in Diverse Environments*. IGI Global (2014)
13. Marsh, S.: *Formalising Trust as a Computational Concept*. Ph.D. thesis, University of Stirling (1994). <http://www.stephenmarsh.ca/pubs/Trust/PhD/Trust.pdf>
14. Marsh, S., Briggs, P.: Examining trust, forgiveness and regret as computational concepts. In: Golbeck, J. (ed.) *Computing with Social Trust*. Human Computer Interaction Series, pp. 9–44. Springer, Heidelberg (2009). Chap. 2

15. Marsh, S., Briggs, P., El-Khatib, K., Esfandiari, B., Stewart, J.A.: Defining, investigating device comfort. *Inf. Media Technol.* **6**(3), 914–935 (2011)
16. Marsh, S., Dibben, M.R.: Trust, untrust, distrust and mistrust – An exploration of the dark(er) side. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) *iTrust 2005*. LNCS, vol. 3477, pp. 17–33. Springer, Heidelberg (2005)
17. Mayorga, R.V., Perlovsky, L.I.: *Toward Artificial Sapience: Principles and Methods for Wise Systems*. Springer, Heidelberg (2007)
18. McKenna, B.: The multi-dimensional character of wisdom. In: Thompson, M.J., Bevan, D. (eds.) *Wise Management in Organisational Complexity*, pp. 13–33. Palgrave MacMillan, London (2013)
19. McKenna, B., Rooney, D.: Wise leadership and the capacity for ontological acuity. *Manage. Commun. Q.* **21**, 537–546 (2008)
20. Padoa, C., Schneider, D., de Souza, J.M., Medeiros, P.J.: Investigating social curation websites: a crowd computing perspective. In: 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 253–258. IEEE (2015)
21. Prensky, M.: *Brain Gain: Technology and the Quest for Digital Wisdom*. Palgrave MacMillan, New York (2012)
22. Rugai, N.: *Computational Epistemology: From Reality to Wisdom*. Lulu Press, Brooklyn (2012)
23. Sevilla, D.C.: The quest for artificial wisdom. *AI Soc.* **28**, 199–207 (2011)
24. Simperl, E.: How to use crowdsourcing effectively: Guidelines and examples. *Liber Q.* **25**(1), 18–39 (2015)
25. Suarez, J.F.: *Wise by Design: A Wisdom-Based Framework for Innovation and Organizational Design and its Potential Application in the Future of Higher Education*. Ph. D. thesis, Antioch University, Dissertations & Theses. Paper 131 (2014). <http://aura.antioch.edu/etds/131>
26. Takahashi, M., Bordia, P.: The concept of wisdom: A cross-cultural comparison. *Int. J. Psychol.* **35**(1), 1–9 (2000)
27. Takahashi, M.: Toward a culturally inclusive understanding of wisdom: Historical roots in the east and west. *Int. J. Aging Hum. Dev.* **51**(3), 217–230 (2000)

Trust It or Not? An Empirical Study of Rating Mechanism and Its Impact on Smartphone Malware Propagation

Wenjuan Li¹(✉), Lijun Jiang¹, Weizhi Meng², and Lam-For Kwok¹

¹ Department of Computer Science,
City University of Hong Kong, Kowloon Tong, Hong Kong
wenjuan.li@my.cityu.edu.hk

² Infocomm Security Department,
Institute for Infocomm Research, Singapore City, Singapore

Abstract. Malicious applications (malware) have attracted much attention from both academia and industry. Thanks to this, common users start to install anti-malware tools to help protect their phones. However, we notice that attackers can still take advantage of some existing mechanisms to induce users to download malware and bypass anti-malware software. In this paper, we focus on the app rating mechanism on smartphones and aim to evaluate its impact on malware propagation. More specifically, we investigate how this mechanism can be maliciously used to leverage the trust levels of users and achieve particular goals (i.e., inducing users to download malware). In the evaluation, we develop a malicious rating system and conduct a study with over 400 participants. Our results indicate that such rating mechanism can affect users' trust on app download and can be utilized to propagate malware.

Keywords: Malicious applications · Anti-malware software · Rating mechanism · Smartphone security · User trust and awareness

1 Introduction

Thanks to the significant portability and the availability of mobile applications, smartphones have quickly become one prevalent computing platform. According to a report from the International Data Corporation (IDC), most existing markets will have a robust growth annually and the worldwide smartphone shipment volumes are forecast to reach 1.9 billion units by 2019 [2]. Since the popularity of smartphones increases, security challenges and threats also become a big concern on this platform.

Malicious applications (malware) are one of these challenges. There were more than 317 million new pieces of malware created in 2014, meaning nearly one million new threats were released into the wild each day [9]. Due to the large

Weizhi Meng was previously known as Yuxin Meng.

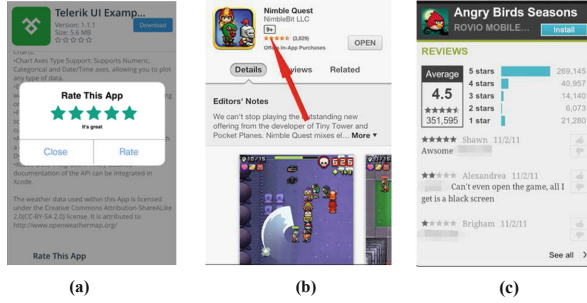


Fig. 1. App rating mechanism: (a) rating action, (b) overall rating score and (c) rating score details and users' comments.

news reports and propaganda, more and more users are willing to install one kind of anti-malware software to safeguard their smartphones. In the literature, many efforts have been made through developing various advanced anti-malware techniques. Differently, our interest in this work was motivated by the fact that attackers may invade users' phones through reducing user awareness, even if an anti-malware software is pre-installed.

In this work, we take *app rating mechanism* on phones as a case and conduct an empirical study to investigate its impact on users' trust and malware spread. In particular, we develop a *malicious rating mechanism* that can leverage the rating scores and comments of target apps. Our purpose is to investigate whether users can be induced to download those malicious apps under the rating system. If users download and install these apps, a self-defined message will be sent to our server and a high rating score will automatically give to those apps in turn. Totally, over 400 users are involved in our study and provide their feedback about their attitude and behaviors. The study results are evaluated based on statistical data and users' feedback. To sum up, the results reveal that attackers can make use of rating mechanisms to reduce user awareness and increase their trust levels on app download, which may greatly degrade the effectiveness of anti-malware tools and cause malware propagation on smartphones.

The remainder of this paper is organized as follows. Section 2 introduces the typical rating mechanism and Sect. 3 describes the developed malicious rating system that was used in the study. Then, Sect. 4 illustrates our study methodologies and analyzes the collected results, and Sect. 5 discusses related studies. Finally, we conclude our work in Sect. 6.

2 Rating Mechanism

This rating mechanism is a user feedback channel, attempting to encourage users to share their experience for downloaded apps. The major purpose of this mechanism is to promote the high rating apps and to allow users to share their experience associated with their used apps. Several examples of this mechanism

are depicted in Fig. 1. Specifically, Fig. 1(a) shows the rating action (i.e., giving a rating score to an app), Fig. 1(b) describes the overall rating score for an app and Fig. 1(c) presents the rating score and users' comments. To summarize, there are three main features of a typical rating mechanism: (1) the rating score usually ranges from 1 to 5, where 5 indicates the highest score (satisfied); (2) the overall rating score is shown on the downloading page of an app and (3) for each app, the details of rating score (e.g., votes for each score) and users' comments are available to all users.

3 Malicious Rating System

As the app rating mechanism may play an important role in users' downloading choice, it is possible for attackers to utilize it to reduce user awareness and increase their trust on particular apps. To validate this, we develop a malicious rating system and its high-level architecture is depicted in Fig. 2. In real-world cases, this alternative market can be popular under some scenarios. For example, an app is not free in official markets but is available in an alternative market (e.g., Anzhi market [1]).

3.1 Popular App Category

To simulate a typical app market, it is expected to provide more popular apps that are likely to be downloaded by users. To achieve this goal, we conduct a survey about popular app categories with 729 participants. The concrete question is: *what is your most frequently downloaded app category*. Each participant can only vote one category. The voting results are shown in Fig. 3.

It is found that up to 302 participants (with a rate of 41.4%) are frequently to download *entertainment apps* (e.g., games). There are 183 participants (about 25.1%) are often to download *tool apps*. In contrast, few participants (less than 2%) download *puzzle* and *education apps* in their spare time. Based on these results, we deploy apps in the market with a similar distribution (i.e., entertainment apps are the most).

3.2 Market Settings and Malicious Rating

In this section, we discuss how to construct the app market in detail and how the malicious rating system works.

- **Market construction.** According to the app distribution in Fig. 3, our app market is configured to contain a total of 1405 apps, where 603 of them are entertainment apps, 354 of them are tool apps and the remaining apps belong to other relevant categories. Users can connect to the market through a web link, search and download apps to their mobile phones. The rating scores (the highest score is 5) and users' comments would be shown on each app-downloading page.

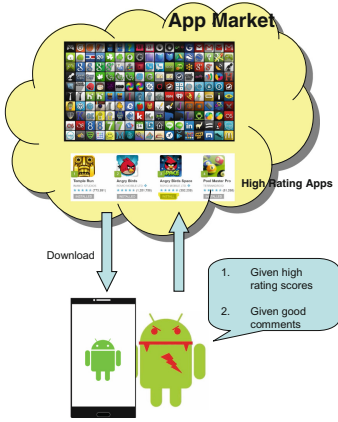


Fig. 2. The high-level architecture of our malicious rating system.

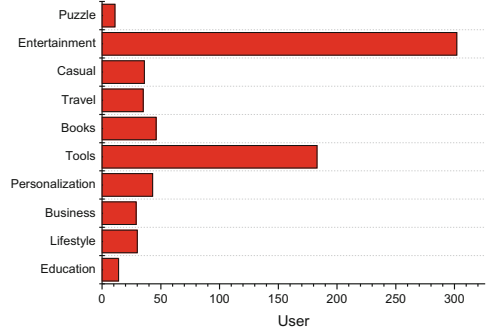


Fig. 3. A survey of popular app category.

- **Malicious rating system.** To deploy those malware to our market, we camouflaged 500 entertainment and 300 tool apps using such malware. Moreover, to explore the effect of rating scores on users' choice, 300 entertainment apps were given a high rating score between 4.5 and 5, while the other entertainment apps were given a low rating score between 2 and 2.5. It is the same for other app categories, in which only half of them were given a high score. After users' installation of our malicious apps, a message will be sent to the server. Afterwards, the server will give the highest score (5 score) to that app and generate a corresponding benign comment.

4 Our Study

Due to security reasons, it is not feasible to use an existing app market in our study directly. In this case, we developed a web-based app market (named as *Popstar market*) on our self-maintained server, in which its structure and workflow are similar to the existing Anzhi market [1].

4.1 Study Methodology

In the study, most participants were recruited from a university environment, since students are one of the main body of smartphone users. Before the study, we got approvals from the university and security office, so that we can use an online system to recruit participants. All study deployment and processing were not relevant to the university environment. More specifically, we mainly conducted two case studies: *in-lab study* and *out-lab study*.

- **In-lab study.** Participants were recruited online, where a total of 317 students were willing to attend this study. None of them were from security-related

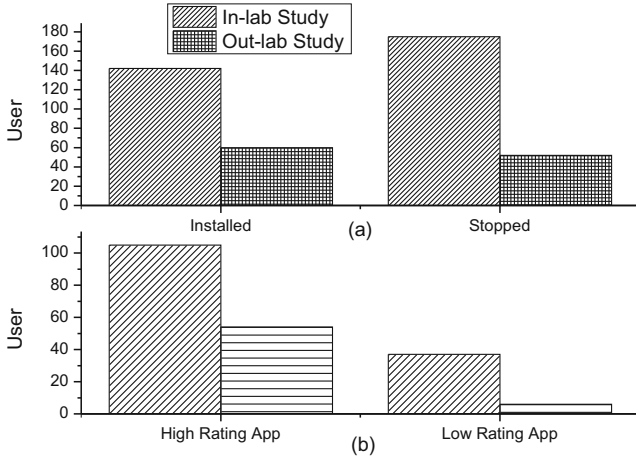


Fig. 4. (a) Infection rates for in-lab and out-lab conditions, and (b) Installed high rating apps versus low rating apps.

majors. All participants were invited to our lab and were given an Android phone. The phone was pre-installed an antivirus tool that can make an alert when the user encounters and installs the developed malware. This design aims to explore whether users will ignore such kind of alerts. In addition, before the study, we explained our goal to all participants, and they were required to download up to 15 apps. After the study, all participants were given a questionnaire form to provide their feedback. A gift card \$10 will be given to each participant as incentive.

- **Out-lab study.** In order to simulate a real scenario, we conducted another case study, where users did not need to come to the lab environment to complete their tasks. In fact, all participants should use their own phones to finish the study. For this purpose, 112 new students were recruited online. None of them were from security-related majors. Before the study, we introduced our objectives to all participants. Similar to our tasks in the in-lab condition, all participants were asked to download 15 apps from our market. After the study, all participants were invited to give feedback on a questionnaire form. Finally, 109 participants were consent to give feedback and approved us to use the statistical data. In this case, we only analyze the data from these 109 participants and all of them were given a \$10 gift card as incentive. The whole study took three weeks to finish.

4.2 Statistical Results

According to our system design, as long as one participant successfully installed the malicious app, our server can receive a message for that app. This can help compute statistics relevant to users' choice and actions during the app download.

Table 1. User feedback during the study

Questions/users	<i>In-lab</i>	<i>Out-lab</i>
Q1. I prefer to download my interested apps	188	96
Q2. I prefer to download high rating apps	226	100
Q3. Did you check others' comments before downloading	242	99
Q4. Did you notice any app requiring to disable antivirus software	270	97
Q5. Did you stop any antivirus software before downloading apps	142	60
Q6. Were you aware of any risk of downloading apps in the market	171	55

After the study, we calculate the statistical results based on the received messages from the server. The infection rates and the numbers of installed high and low rating apps are shown in Fig. 4.

- **In-lab study.** Figure 4(a) shows that up to 142 participants suffered from the malicious rating system and installed the malware, in which the infected rate is 44.8%. In the study, it is observed that those infected participants have ignored the pop-up alert and disabled the pre-installed anti-malware software. To look closer to the installed number of different rating apps, Fig. 4(b) shows that those participants are very likely to continue installing an application with a high rating score, instead of installing a low rating app. The figure presents that nearly 74% installed malicious applications are high rating apps, while only 26% apps with low rating scores.
- **Out-lab study.** The main purpose of this study is to investigate users' behavior in a practical environment. Under the out-lab condition, users can use their own phones to search and download apps from our market. Similarly, the infection rate and installed app numbers are shown in Fig. 4. Figure 4(a) shows that up to 60 participants continued installing the malicious apps on their own phones so that the infected rate is 53.6%. The infection rate is a bit higher than that of the in-lab study. According to Fig. 4(b), participants were mostly willing to continue the installation of high-rating apps, where 90% successful cases came from high-rating apps.

4.3 User Feedback

In this part, we analyze the collected user feedback regarding their choice, awareness and trust levels during the app download. Some key questions and user feedback from both case studies are summarized in Table 1.

App Download Choice. In Table 1, the first question reflects that 59.3% participants selected to download high-rating apps in the lab environment, but more than 85% participants did so outside the lab (in the second user study). In contrast, in the out-lab study (with their own phones), users prefer to start downloading their interested apps. The numbers of this question indicate that users usually begin downloading an app according to their interests, even under a constrained condition.

In Table 1, the second question shows that 226 out of 317 participants (about 71.3%) were likely to download high-rating apps under the in-lab condition, while 100 out of 109 participants (about 91.7%) prefer to download high-rating apps under the out-lab condition. The latter is much higher than the former due to environmental factors, whereas both rates verify that users are willing to install high-rating apps. They generally believe that high-rating apps are more secure than those low rating ones. For example, if there are several app versions, they are more likely to download the version that has a higher score.

User Trust. The third question shows that up to 76.3% and 90.8% participants in respective condition will check others' comments before they download an app. In our interview, it is found that users' comments have a high impact if the app download or installation encounters some issues. For example, when the app installation is alerted by an antivirus, users will check others comments to confirm the situation. If the comments are good enough, users may decide to ignore the alert and continue the installation.

Regarding the fourth question, most participants (over 85%) notice that some apps require to disable anti-malware software before downloading & installing the apps. The fifth question shows that nearly half participants would follow the instructions to disable the anti-malware software and continue the installation. In our interview, most infected participants considered that this may be a common case for some entertainment and tool apps, especially in a new market, that false alarms often occur.

User Awareness. The last question shows that only about half participants (53.9% and 50.4% for each condition) are really aware of any risk in downloading apps from our market. Most participants considered that those apps, especially high-rating apps, should be benign and at least not harmful, since the rating scores and comments are quite good from others.

Discussions. Overall, based on the feedback, it is validated that rating scores and comments can greatly affect users' trust on app download. That is, the rating mechanism can impact users' attitude in downloading an app from a market. Therefore, through proper camouflage, attackers can utilize such rating mechanism to induce users to disable anti-malware tools and continue downloading & installing particular apps. This situation opens a hole for attackers to spread malware even if users have pre-installed antivirus software.

5 Related Work

As smartphones have become a major target for attackers, various research studies have focused on the detection of malware from market [8, 10]. There are also some studies discussing recommender and rating systems [3, 7]. Different from those studies, in this work, we target on app rating mechanism and attempt to evaluate its impact on users' trust on app download and malware propagation on smartphones. From this aspect, to the best of our knowledge, our work is the first study in the literature to investigate this topic.

Our study reveals that users' trust can be greatly affected by a malicious rating system and be induced to download malicious apps, resulting in malware propagation. Even worse, it is worth noting that such malicious rating system can collaborate with existing advanced malware techniques to achieve an even larger impact (i.e., stealing users' sensitive information and data). There is a line of research studies on applying various attacks to infer users' private information and data on smartphones, including side channel attacks [5] and physical access attacks like charging attacks [4, 6].

6 Conclusion

Different from other studies on rating systems, in this paper, we focus on app rating mechanism on smartphones and aim to evaluate its impact on users' trust and malware propagation. We have two specific questions: whether users can be induced to download malicious apps, and whether the rating systems can affect users' trust on app download. Our results indicate that users' trust can be greatly affected by such system by manipulating high rating scores and good comments. By taking advantage of this system, attackers can propagate malware and bypass antivirus tools. Our research attempts to raise more attention for malware research community on user-centric solutions.

References

1. Anzhi Market. www.anzhi.com/
2. Global Smartphone Growth Expected to Slow to 11.3% in 2015. <http://www.idc.com/getdoc.jsp?containerId=prUS25641615>. Accessed June 2015
3. Jøsang, A.: Robustness of trust and reputation systems: does it matter? In: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (eds.) IFIPTM 2012. IFIP AICT, vol. 374, pp. 253–262. Springer, Heidelberg (2012)
4. Meng, W., Lee, W.H., Murali, S.R., Krishnan, S.P.T.: Charging me and I know your secrets! Towards juice filming attacks on smartphones. In: Proceedings of ACM CPSS, pp. 89–98 (2015)
5. Meng, W., Wong, D.S., Furnell, S., Zhou, J.: Surveying the development of biometric user authentication on mobile phones. *IEEE Commun. Surv. Tutorials* **17**(3), 1268–1293 (2015)
6. Meng, W., Lee, W.H., Krishnan, S.P.T.: A framework for large-scale collection of information from smartphone users based on juice filming attacks. In: The Singapore Cyber Security R&D Conference (SG-CRC), pp. 99–106 (2016)
7. Muller, T., Liu, Y., Mauw, S., Zhang, J.: On robustness of trust systems. In: Zhou, J., Gal-Oz, N., Zhang, J., Gudes, E. (eds.) IFIPTM 2014. IFIP AICT, vol. 430, pp. 44–60. Springer, Heidelberg (2014)
8. Peng, S., Yu, S., Yang, A.: Smartphone malware and its propagation modeling: a survey. *IEEE Commun. Surv. Tutorials* **16**(2), 925–941 (2014)
9. Symantec. Internet Security Threat Report, vol. 20 (2015). http://www.symantec.com/security_response/publications/threatreport.jsp. Accessed June 2015
10. Teuffl, P., Ferk, M., Fitzek, A., Hein, D., Kraxberger, S., Orthacker, C.: Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play). *Secur. Commun. Netw.* **9**(5), 389–419 (2016)

Towards Behavioural Computer Science

Christian Johansen¹(✉), Tore Pedersen², and Audun Jøsang¹

¹ Department of Informatics, University of Oslo, Oslo, Norway

{[cristi,josang](mailto:cristi,josang@ifi.uio.no)}@ifi.uio.no

² Center for Intelligence Studies,

Norwegian Defence Intelligence School, Oslo, Norway

tore.pedersen@feh.mil.no

Abstract. The rapidly increasing pervasiveness and integration of computers in human and animal society calls for a broad discipline under which this development can be studied. We argue that to design and use technology one needs to develop and use models of humans/animals and machines in all their aspects, including cognitive and memory models, but also social influence and (possibly artificial) emotions. We call this discipline Behavioural Computer Science (BCS), and propose that BCS models combine (models of) the behaviour of humans/animals with that of machines when designing ICT systems. Incorporating empirical evidence for actual human behaviour instead of relying on assumptions about rational behaviour is an important shift that we argue for. We provide a few directions for approaching this challenge, focusing on modelling of human behaviour when interacting with computer systems.

1 Introduction

The marriage of ubiquitous computing and AI opens up an environment where complex autonomous systems are heavily involved in the living and working environments of humans, often in a seamless fashion. Not only must humans relate to intelligent machines, but the same machines must relate to humans and to other intelligent machines.

Our ethical compass should guide us to build intelligent machines that have desirable traits, whatever that might be. In order to achieve this goal it is essential that we understand how humans actually behave in interactions with intelligent machines. For example, what are the criteria for trusting an intelligent machine for which the intelligent behaviour *a priori* is unknown. Also, how can an intelligent machine trust humans with whom it interacts. Finally, how can intelligent machines trust each other. From a security point of view, the most

A long version of this paper is available as the technical report [13].

C. Johansen was partially supported by the project [OffPAD](#) with number E!8324 part of the [Eurostars](#) program funded by the [EUREKA](#) and European Community.

T. Pedersen and A. Jøsang were partially supported by the project [Oslo Analytics](#) funded by the [IKTPLUSS](#) program of the [Norwegian Research Council](#).

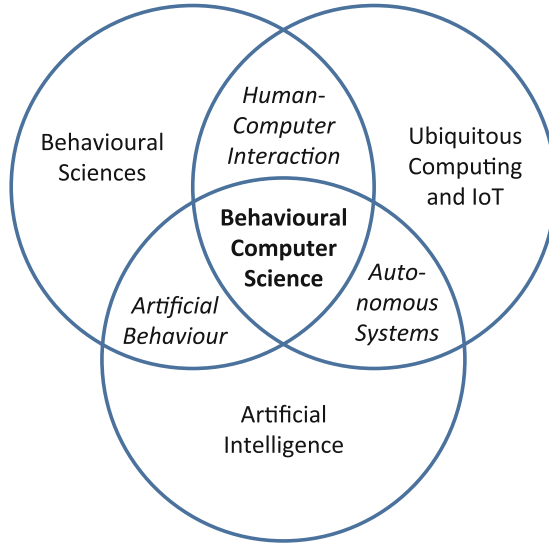


Fig. 1. Conceptual definition of behavioural computer science

serious vulnerabilities are no longer found in the systems but in the humans who operate the systems. In a sense, it is no longer a question of whether people can trust their systems, but whether systems can trust their human masters.

These are daunting challenges in the brave new world of intelligent ubiquitous computing and cyberphysical infrastructure. Three important fields of scientific study are fundamental to understanding and designing this infrastructure:

Behavioural Sciences giving scientific, empirical, evidence-based, and descriptive models for how people actually make judgements and decisions, as opposed to the traditional, rational and normative approaches that describe how people should ideally behave. Examples of behavioural sciences include psychology, psychobiology, criminology and cognitive science.

Ubiquitous Computing and IoT as the new paradigm in computer science where computing is made to appear everywhere, in various forms and everyday objects such as a fridge or a pair of glasses. Thus appear new forms of user interactions with such systems. The underlying technologies include Internet, advanced middlewares, sensors, microprocessors, new I/O and user interfaces. The IoT is the connected aspect of ubiquitous computing.

Artificial Intelligence studying how to create computers and computer software that are capable of intelligent behaviour. AI is defined in [23] as “the study and design of intelligent agents”, in which an intelligent agent is a system that perceives its environment and takes actions that maximise its chances of success according to some criteria.

We put these three areas under the same umbrella called “*Behavioural Computer Science*” (abbreviated BCS) and illustrated in Fig. 1. Any outcome of

integrating models from these three areas would be called a *BCS-model*, which should also include aspects of human behaviour. We would like to encourage research focus on the interactions between these three areas. The intersections between any two of these areas represent existing or new research disciplines.

Human-Computer Interaction (HCI) and more recently Interaction Design [24] studies how a technology product and its interface should be developed having the user in focus at all stages. With the advent of ubiquitous computing, the Internet of Things (IoT) and advanced AI, the distinction and interface between computers and humans becomes very blurred.

Models for how humans and intelligent machines interact can be understood in a general and inclusive manner, as any formally or mathematically grounded model used in building IT systems. We can think of probabilistic models, logical and formal models, programming and semantic models, etc. One purpose of using models is to understand and reduce complexity.

Computational trust becomes an aspect of machine learning or heuristics, that in turn will be part of IoT systems and other (semi-)autonomous controllers, or self-* systems. For such autonomous and powerful systems we need to study notions of trust [15], like trust of the user in the system, or of another interacting system or component.

2 Behavioural Aspects of Humans and Technology

When humans interact with technology it is necessary to understand human behaviour in order to capture or foresee possible actions taken by humans. We refer here to an understanding that can be used by machines, thus through models that can be used in forms of computations. If technology and its designers understand the typical tendencies of human cognition, emotion and action, it is easier for the resulting system to take into consideration how people actually behave, and adapt in accordance.

Traditionally we find the Rational Agent Model (e.g., [27]) for explaining human behaviour, which generally adheres to the view that people are rational agents. However, it has been argued that the assumptions of the rational agent are seldom fulfilled, which leads us to focus on the Behavioural Model of Human Agency, as proposed by notable researchers like [10, 18, 28, 31].

The rational agent model implies that people always strive to maximize utility, generally understood as the satisfaction people derive from the consumption of services and goods [20]. If one looks at utility from a psychology perspective, a problem arises because there is more than one definition of utility [18].

Experienced utility is the satisfaction derived in the consumption moment.

Predicted utility (or, alternatively, *expected* utility) is the utility one predicts beforehand that one will experience in the future consumption moment.

Remembered utility is the utility one remembers having experienced in a consumption moment some time ago.

The rational agent model implicitly assumes them to be equal, whereas empirical psychology research has found that these aspects reflect different utilities. *Errors in human behaviour* often stem from the differences between predicted, experienced and remembered utility; e.g. when making judgements at time t_0 about some consumption related moment in the future at t_1 , one often disregards the fact that their current experiences will be different from their expectations.

Rationality assumes that people act strictly logical, in the pursuit of maximized utility. In consequence, conditions are assumed to be certain, with humans having unlimited access to all information and also capable of analysing the relevant information needed to make a judgement, as well as calculating the outcome of every combination of informational components. However, behavioural scientists [18] questioned the explanatory powers of the rational agent model, because they could not make their empirical data fit the rational agent model. A new view, supported by empirical data emerged, showing that people's judgement errors were not at all random, but in fact systematic; people tended to make the same kinds of misjudgements as others did, and misjudgments made today are the same as those made yesterday. Moreover, *human errors* appear also as a consequence of making judgements in conditions under uncertainty, i.e., when the requirements of the rational agent model cannot be fulfilled.

One universal finding in this new avenue of research is that there are two fundamentally different systems of cognitive processing [17, 29]:

System 1: Intuitive Thinking, is associative, effortless, emotion-influenced, automatic, and thus often operating without conscious awareness;

System 2: Analytic Thinking, is analytic, effortful, not influenced by emotions, sequential, controlled and thus operating with conscious awareness.

Because Intuitive Thinking is effortless and automatic, people have a tendency to rely heavily on this cognition mode in most everyday activities – where we automatically know how to judge, behave and decide. The problem is when this automatic mode of thinking is applied in situations where we do not have enough knowledge or experience. A failure to activate Analytic Thinking in these situations may lead to systematic errors, also labelled **biased judgements**.

Another finding from behavioural sciences that is relevant to BCS is four psychological mechanisms (also called *heuristics*) that are mostly responsible for the human tendency to make unwarranted swift judgements [10]. These belong to Intuitive Thinking and lead to biases in situations where we are uncertain.

The availability heuristic explains how people make judgements based on what is easily retrievable from memory, or simply what comes easily to mind.

The representativeness heuristic describes how people make a judgement based on how much the instance or the problem in front of them is perceived as similar to another known instance or problem. If the degree of perceived similarity is large enough, people will easily make incorrect judgements.

The anchoring and adjustment heuristic implies that people – under conditions of uncertainty – without conscious awareness will establish an “anchor”,

and from this anchor adjust their judgement, often in the “right” direction, although not to the point of accuracy. If you are in a condition of total uncertainty, even non-relevant information that you have either been primed with, or that is easily accessible from memory, can serve as an anchor.

The affect heuristic explains how the current affective state may influence human judgements, e.g., when in a positive mood, one may be more easily susceptible to deception and manipulation.

To counteract the tendency towards the Intuitive Thinking, one possible intervention is to “slow” people’s actions down, thereby making them employ System 2-thinking. The message that we get when trying to delete a file, saying “Are you sure you want to delete this file?” is an example of such an intervention.

A spear phishing attack, where one receives a malicious email from an address that resembles that of a known colleague, is difficult to counter because it *activates both the availability heuristic and the representative heuristic*; the user may not have easily accessible information stored in her mind that may suggest that this is an hostile attack (susceptibility to the availability heuristic) and, furthermore, the user recognizes the email address as being from a near colleague (the representative heuristic).

Human choices and human prediction power are very important for interactions with computer systems, e.g. security can be influenced by poor predictions about the possibilities of attacks, and attack surface can be wrongly diminished in the mind of the human, whereas wrong choices can incur safety problems. In [18] it is argued that it is difficult for a human to make accurate predictions about a situation or an experience (e.g., sentiment, preference, disposition) when the future forecasting time point t_1 is rather distant from the current time point t_0 on which the same experience is evaluated. The more distant this time point is, the more inaccurate the prediction (and thus the choice) will be.

3 Modelling for Behavioural Computer Science

We anchor our thoughts using concepts from a model introduced in [3], which we call “*the Bella-Coles-Kemp model*” and abbreviate as *BCK model*. More details not necessarily relevant for this section can be found in [13, Sect. 3] or in [12] where we used the BCK model in the context of security ceremonies. We will call the human *the Self*, which can be *influenced* by the *Society*, e.g., through social-engineering methods. The Self is *expressed* for a particular computer system as a *Persona*, understood as a collection of attributes relevant for a particular system interaction. The Persona is *interacting* with the system through the *User Interface (UI)*, often called *socio-technical protocol*. Socio-technical protocols have been studied in the Human Computer Interaction and related fields [4, 5, 24].

We are interested in how behavioural concepts could be mathematically modelled, and more importantly, how these behavioural models can be coupled and integrated with existing models from computer science. We discuss a few aspects, some related to works from HCI [7, 25] and from cognitive theories [19].

Kahneman and Thaler [17, 18] argued that the circumstances (i.e., the context of the human and of the system) vary between the present t_0 and future t_1 time points. Four large areas of such *varying circumstances* can be identified:

The emotional state of the human, or the **motivational state** of the human might vary when t_0 and t_1 are distant.

The aspects of the choice, of the product, of the experience, that are considered important or are made salient/observable at t_0 , might not be present at t_1 or may be difficult to experience or observe at this later time point.

Memory of similar choices or experiences is important. If the memory is biased then the current choice and prediction for the future will be biased. Tests of memory manipulation have been made [16] and one observation is summarized as the *Peak/End Rule*, as opposed to the common belief that the monotonicity of the experience counts. Humans recall the experiences of the peak emotions or of the end of the episode.

Affective forecasting [21, 33] – the process of predicting future emotions – explains how when focusing on some aspect for making a decision, this aspect may inappropriately be perceived as more important at the time of (prediction and) decision than it normally will be at the time of experience.

We will work with a notion of “States” and changes between states (which we call “Transitions”). Modelling an *emotional or motivational state* is not trivial, so let us look at the *changes* between states first. We have already discussed about “*temporal changes*”, i.e., changes that happen because of passage of time. These we can consider in two fashions:

gradual/continuous change in emotion or motivation happens over time, (e.g., modelled with time derivatives, in the style of physics); or

discrete changes where we jump suddenly from one value to a completely different value (e.g., think of motivation which can gradually decrease until it reaches a threshold where it is suddenly completely forgotten).

For modelling *emotions* (as needed for *affective forecasting* and many aspect of the Self) we start from the two concepts related to the *impact bias* [33]: the *strength* (or *intensity*) of an emotion and the *duration*. Both can be quantified and included in a *quantified model of emotions*. Other temporal notions different than durations could be needed like *futures* or order *before/after*, for which there are well established models in computer science, e.g. temporal logics [2, 30].

Also influencing the Self are **events**, since *emotions are relative to events*. Events can be considered instantaneous and modelled as *transitions* labelled by the event name, because an event changes the state in some way, e.g., changes the memory of the Self, or attributes of the context as well as of the Self.

These concepts contribute to defining *models for the predicted and the remembered utilities*, as well as their correlation with that of the experienced utility.

For *modelling a State* we start by including the *aspects* of interest for the situation under study. Aspects could be modelled as logical variable that are true or false in some state, because they are either considered or not considered

(i.e., observable/salient or not). The expressiveness of the logic to be used would be dependent on what aspects we are interested in; but we can start by working with predicate logic. Depending on the system being developed, we encourage to choose the most suited logic, e.g.: the SAL languages and tools which have been nicely used to describe the cognitive architecture of [26, Sect. 2]; or one can use higher-order dynamic logic [11, Chap. 3] for more complex structures.

The relation between the Self and the Persona can be seen as a simplification (or projection). The projection operation is done on a subset of the variables that make up the State of the Self, thus resulting in the state of the Persona. This projection would retain only those aspects that are relevant in the respective context, i.e., in the context of the computer system being studied. This means that the projection operation should also be related to the model of the UI.

Besides the simplification relation we need to understand the interactions between the Self and the Persona. We can see **two interaction directions**:

- from the Persona to the Self** i.e., to the user with all the experiences, sensors, memory, thinking systems, heuristics, etc.; and
- from the Self to the Persona** i.e., to a simplified view of the user, specifically made for the UI and the system being studied.

Since a Persona is an abstraction of the human relevant for the interaction with a specific UI, then through the Persona we can see stimuli from the UI going to the Self, and influencing it. Therefore, the first communication direction can be seen as communications coming from the UI but *filtered through the Persona*.

The second direction considers actions of *expression* (e.g., described by [7, 26]) that the Self makes out of the thoughts, reasoning, intuition, past experiences and memory models, filtered by the Persona and directed towards the UI.

Such interactions would be studied empirically, looking at the Self and Personas. A model *starts from general assumptions*, incorporated as prior probabilities. For a specific system, with a specific Persona defined, the model would constantly be updated by learning from the empirical studies and evidences.

Because we use empirical evidences we need to introduce a notion of *uncertainty about the probabilities* that the studies reveal. Therefore, models of *subjective logic* [14] could be useful for expressing things like: “The level of uncertainty about this value given by this empirical study is the following”.

One would then be interested in applying standard analysis techniques like model-checking over these new models with uncertainty. This would allow to:

- Identify ways to protect the Self from malicious inputs and manipulation from the UI through the Persona.
- Identify ways to protect the Self from social-engineering attacks.

One type of protective methods are *debiasing techniques* [22], useful for countering biases caused by the focusing illusion. A BCS system could implement, part of the UI or the security protocol, features meant to manipulate the User in such a way that she would be prepared for a possible attack. Such features could involve: recollections, so that the same aspects of t_1 (now) are as in t_0 (the time point when the User has probably been trained in using the system).

4 Further Work

We argued that concepts and findings from behavioural sciences can be translated into models useful for computer science. Such models could be used for analysing the BCS-systems using techniques such as automated model checking [2]. Moreover, behavioural models and related modelling languages can be used by system developers when making new BCS-systems to also consider the human interacting with the system. We can already see promising results in this direction from using formal methods to analyse HAI systems [5] or human related security breaches [26].

Consider three examples where the behavioural approach to explaining human judgment has successfully enriched an existing academic discipline:

Behavioural Economics focusing on how people actually behave in economic contexts, as opposed to how they should ideally behave (e.g., [17,18]), has been a fruitful addition to Economics;

Behavioural Game Theory focusing on how people actually behave in formal games, as opposed to how they should ideally behave (e.g., [6]), has enriched traditional Game Theory; and

Behavioural Transportation Research focusing on how people actually make choices in transportation and travel contexts, as opposed to how they are assumed to behave (e.g., [9,21]), has been a fruitful addition to the traditionally rationalistic field of Transportation Research.

Our opinion is that Behavioural Computer Science can be one more fruitful collaboration between behavioural models and computer models.

Consider two examples of emerging fields which can be seen as part of BCS.

Security ceremonies propose to involve the human aspect when designing and analysing security protocols [8]. A few works have studied the human aspect of security breaches [26,32]. An example is phishing e-mails where we argue that cognitive models and models of social influence can give insights into how to build e-mail systems that can counter more effectively such targeted, well-crafted, malicious e-mails.

Ambient assisted living [1] is one application of IoT that is most closely interacting with humans. Such systems need to learn patterns of behaviour, distinguishing them among several occupants, adapt to temporary changes in behaviour, as well as interact and take control requests from the humans.

References

1. Augusto, J., Huch, M., Kameas, A., Maitland, J., McCullagh, P., Roberts, J., Sixsmith, A., Wichert, R. (eds.): Handbook of Ambient Assisted Living. IOS Press (2012)
2. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press, Cambridge (2008)

3. Bella, G., Coles-Kemp, L.: Layered analysis of security ceremonies. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012. IFIP AICT, vol. 376, pp. 273–286. Springer, Heidelberg (2012)
4. Bevan, N.: International standards for HCI and usability. *Int. J. Hum.-Comput. Stud.* **55**(4), 533–552 (2001)
5. Bolton, M., Bass, E., Siminicéanu, R.: Using formal verification to evaluate human-automation interaction. *IEEE Trans. Sys. Man Cybern.* **43**(3), 488–503 (2013)
6. Camerer, C.F.: *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, Princeton (2003)
7. Curzon, P., Rukšėnas, R., Blandford, A.: An approach to formal verification of human-computer interaction. *Form. Aspects Comput.* **19**(4), 513–550 (2007)
8. Ellison, C.: Ceremony design and analysis. Cryptology ePrint Archive report 2007/399 (2007)
9. Gärling, T., Ettema, D., Friman, M. (eds.): *Handbook of Sustainable Travel*. Springer, Dordrecht (2014)
10. Gilovich, T., Griffin, D., Kahneman, D. (eds.): *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press, New York (2002)
11. Harel, D., Tiuryn, J., Kozen, D.: *Dynamic Logic*. MIT Press, Cambridge (2000)
12. Johansen, C., Jøsang, A.: Probabilistic modelling of humans in security ceremonies. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/SETOP/QASA 2014. LNCS, vol. 8872, pp. 277–292. Springer, Heidelberg (2015)
13. Johansen, C., Pedersen, T., Jøsang, A.: Reflections on behavioural computer science. Technical report 452, Department of Informatics, University of Oslo, April 2016
14. Jøsang, A.: A logic for uncertain probabilities. *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* **9**(3), 279–311 (2001)
15. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **43**(2), 618–644 (2007)
16. Kahneman, D.: Evaluation by moments, past and future. In: Kahneman, D., Tversky, A. (eds.) *Choices, Values and Frames*, pp. 693–708. Cambridge University Press, New York (2000)
17. Kahneman, D.: A perspective on judgment and choice: mapping bounded rationality. *Am. Psychol.* **58**, 697–720 (2003)
18. Kahneman, D., Thaler, R.H.: Anomalies: utility maximization and experienced utility. *Journal Econ. Perspect.* **20**(1), 221–234 (2006)
19. Newell, A.: *Unified Theories of Cognition*. Harvard University Press, Cambridge (1990)
20. Oliver, R.L.: *Satisfaction: A Behavioral Perspective on Consumer*. Sharpe (2010)
21. Pedersen, T., Friman, M., Kristensson, P.: Affective forecasting: predicting and experiencing satisfaction with public transportation. *J. Appl. Soc. Psychol.* **41**(8), 1926–1946 (2011)
22. Pedersen, T., Kristensson, P., Friman, M.: Counteracting the focusing illusion: effects of defocusing on car users predicted satisfaction with public transport. *J. Environ. Psychol.* **32**(1), 30–36 (2012)
23. Poole, D., Mackworth, A.: *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, New York (2010)
24. Rogers, Y., Sharp, H., Preece, J.: *Interaction Design: Beyond Human-Computer Interaction*, 3rd edn. Wiley, New York (2011)
25. Rukšėnas, R., Back, J., Curzon, P., Blandford, A.: Verification-guided modelling of salience and cognitive load. *Formal Asp. Comput.* **21**(6), 541–569 (2009)

26. Ruksenas, R., Curzon, P., Blandford, A.: Modelling and analysing cognitive causes of security breaches. *Innov. Syst. Softw. Eng.* **4**(2), 143–160 (2008)
27. Simon, H.A.: *Reason in Human Affairs*. Stanford University Press, Stanford (1983)
28. Simon, H.A.: *Models of Bounded Rationality: Empirically Grounded Economic Reason*. MIT Press, Cambridge (1997)
29. Sloman, S.A.: Two systems of reasoning. In: Gilovich, T., Griffin, D., Kahneman, D. (eds.) *Heuristics and Biases: The Psychology of Intuitive Judgment*, pp. 379–396. Cambridge University Press, New York (2002)
30. Stirling, C.: *Modal and Temporal Properties of Processes*. Springer, Heidelberg (2001)
31. Thaler, R.H., Sunstein, C.R.: *Nudge: Improving Decisions about Health, Wealth, and Happiness*. Yale University Press, New Haven (2008)
32. West, R.: The psychology of security. *Commun. ACM* **51**(4), 34–40 (2008)
33. Wilson, T.D., Gilbert, D.T.: Affective forecasting. In: *Advances in Experimental Social Psychology*, vol. 35, pp. 345–411. Academic Press (2003)

Improving Interpretations of Trust Claims

Marc Sel^(✉)

Information Security Group, Royal Holloway,
University of London Egham, Surrey TW20 0EX, UK
Marc.Sel.2013@live.rhul.ac.uk, marc@marcseleu

Abstract. This paper presents an approach to semantic modelling of large-scale trust ecosystems to improve the interpretations of trust claims. The problem of interpreting trust claims is described and relevant types of reasoning are analysed. A model based on *SRIOQ* and OWL Description Logic is proposed. The novel elements are the creation of classes and properties on the basis of legal and regulatory sources that extend existing vocabularies (W3C, Dublin Core), and the use of these classes and properties to create assertions that represent information harvested from on-line information sources. The resulting model allows automated classification via a reasoner, as well as queries that support use cases from various actors. A general approach is presented, as well as results from a prototype implementation based on the European eIDAS and US FICAM trust ecosystems.

1 Introduction

1.1 Context

The digital society will continue to increase its reliance on electronic transactions. Such transactions are conducted between Service Providers and Service Consumers, possibly with the use of intermediaries. Relying on the outcome of a transaction performed via an ICT system, or making a selection which system to use in the first place, forces the user to take a trust decision. While the notion of trust is in widespread use, its meaning varies. For a basic treatment, refer to Gambetta et al. [4], Marsh [9] or Cofta [2].

1.2 Motivation

The motivation for the research described below stems from two observations. First, understanding what a specific trust claim actually means, what it is based on, as well as why it should be considered valid is still hard, and there is often room for different interpretations. This article promotes the view that one should not ‘trust’ but rather take an informed decision on the basis of evidence and reasoning. Second, various actors publish reasonably independent information on other actors in the same ecosystem. For example regulators, central banks, and

business registers provide contextual information that can contribute to verifying a claim. Today's trust models typically include such contextual information only in a limited way. More extensive usage of such information under formal semantics could potentially strengthen the verification of claims because it adds information typically from beyond the control of the actor whose claim is validated. It is to the benefit of honest parties that reliance on a transaction is based on a trust model with semantics and evidence understandable and agreeable to all.

1.3 Research Contributions

This paper researches the type of reasoning that would allow a limitation of interpretation of trust claims. The problem of interpreting trust claims is described. A novel trust modelling approach is proposed, based on a Trust Claim Interpretation model that answers queries resulting from execution of a trust policy validation algorithm. Novel elements are the creation of classes and properties on the basis of legal and regulatory sources that extend existing vocabularies (W3C, Dublin Core), and the use of these classes and properties to create assertions that represent information harvested from on-line information sources. An implementation based on *SROIQ* and OWL Description Logic is presented.

1.4 Paper Outline

The preceding section set the context and motivation, and provided an introduction to the research contribution. Section 2 describes the various types of trust statements addressed, and what existing work has been done in the area. Section 3 describes a new approach to trust modelling, including a novel trust modelling architecture. Section 4 discusses a prototype implementation, based on the choice of *SROIQ* and OWL DL. In Sect. 5 strengths and weaknesses are analysed, as well as areas for further research.

2 Trust, Trust Modelling and Prior Art

2.1 Trust and Trust Modelling

A key part of the development of the electronic society is the introduction of an economy based on electronic transactions and trust. Transactions are conducted between Service Providers and Service Consumers. Trust can be provided by a range of possible mechanisms including, but not limited to, cryptographic protocols and legal or contractual liability. The meaning of 'trust' varies according to the circumstances, and the perspective of the trustee (who is trusted) and the trustor (who is trusting). Trust in cryptographic protocols, often relying on Trusted Third Parties, supports many Internet or closed user group transactions.

Qualifying information or a service in an electronic form as 'trusted' is non-trivial. Many different actors, mechanisms and artefacts collaborate to perform electronic trust transactions. In [12] an informal domain model was introduced. An introduction to eIDAS and FICAM trust models is provided in [13].

2.2 Prior Art

With regard to trust, much research has been conducted to represent real world information and use it as the basis for decisions. A trust calculus for PKI and identity management is proposed by Huang and Nicol, [7]. Measuring and computing trust using subjective logic has been studied by Josang [8]. Hartig defined a trust-aware extension to SPARQL [5]. The Web Of Trust (WOT) project¹ defined an basic RDF vocabulary to facilitate the use of Public Key Cryptography. Shekarpour and Katebi reviewed trust calculation and models of trust rating, and proposed algorithms for propagation and aggregation of trust [14]. A formal notion of trust to enable reasoning about security properties is proposed by Fuchs, Gürgens and Rudolph [3].

3 Trust Modelling: A New Approach

3.1 Outline

A model for reduction of interpretations of trust claims is proposed, combining mathematical modelling with harvesting artefacts that include contextual information, followed by reasoning according to a well specified logic. In Fig. 1 the real world is represented by a globe from where two abstractions are derived. The first abstraction is composed of the actors in the left box. The actors' transactions rely on one or more trust models. To validate a particular reliance, a trust policy validation algorithm attempts to satisfy assumptions by issuing trust claim interpretation requests. The second abstraction is the trust claim interpretation (TCI) model, responding to these requests with responses. For this purpose, the TCI model contains a query engine as well as a knowledge base. The knowledge base contains assertions imported from the real world, and its contents is maintained consistent by a reasoning engine.

3.2 Actors and Their Use Cases

Various actors are involved in the model. A Business Service Producer offers electronic business services, which are consumed by Business Service Consumers. Transactions and connectivity between producers and consumers can be protected by services of Trust Service Providers (TSP). The term TSP refers to a broad category of Service Providers including Identity Providers, Certification Authorities, Signature and Validation Service Providers, Time Stamping Authority services, registered electronic delivery services and trusted electronic archiving services. Other information providers offer additional information. They may be independent from the entities they provide information about in varying degrees. A regulator can impose conditions on entities that provide services.

¹ <http://xmlns.com/wot/0.1/>.

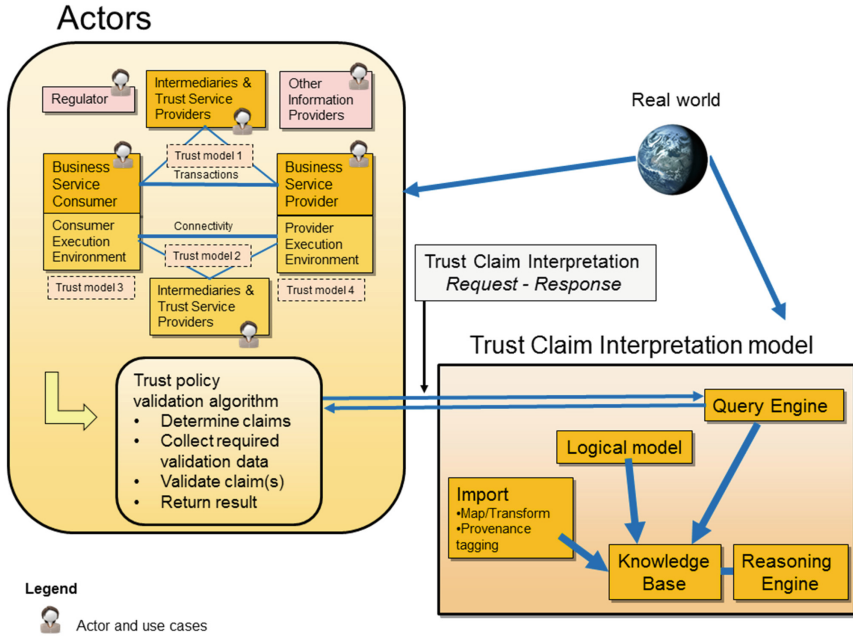


Fig. 1. Overview.

3.3 Trust Models, Trust Policy and Validation

A trust model combines safeguards such as cryptographic protocols, a policy, operational procedures and legal or contractual liability. There might be different trust models deployed at the level of the transactions, the connectivity and the integrity of the operational environment. A trust model is formalised in a trust policy, composed of a set of assumptions which contain a set of claims that need to be satisfied. Reliance can be validated according to a trust policy. The applicable set of assumptions is function of the actor’s use case and the protected artefact. When verification of all claims yields positive evidence that they are satisfied, the assumption is considered satisfied. When all assumptions are satisfied, the trust policy is considered satisfied for that artefact. A generic trust policy is proposed:

- Claimant assumptions, through which a claimant does claim an identity:
 - Claimant functional assumptions that claim the identity for the claimant, specifying the level of assurance required for the authentication thereof, and all supporting evidence related to cryptographic meta data, algorithms and transformation devices.
 - Claimant cryptographic assumptions that address the actual cryptographic validation.
- Claim assumptions, through which a claimant does claim a trust related functionality which is different from identity:

- Claim functional assumptions that address the functionality of the claim (message authentication, electronic signature validation, multi-party signatures validation, etc.), the commitment assumed by the claimant, the level of assurance on timing evidences, and if applicable, the type of legal effect sought by the claim,
- Claim cryptographic assumptions that address the actual cryptographic validation.

The following trust policy validation algorithm is proposed:

1. Determine assumptions and claims in function of protected artefact.
2. Collect supporting validation data. This includes certificates and information on supporting validation data.
3. Validate claimant and claim assumptions. In this step, all applicable assumptions and claims are validated, for which a request/response model is proposed.
4. Return result. The algorithm ends by returning the result to the actor.

3.4 The Trust Claim Interpretation (TCI) Model

The TCI model contains a representation of the real world, derived from normative knowledge and factual assertions. The normative knowledge is derived from authoritative sources such as legislation, regulation, standards and related information. To promote interoperability, the terminology to describe the normative knowledge should be based on existing terms and vocabularies, extended where necessary. Factual assertions are derived from on-line information including published meta data. The import functionality will map and if necessary transform the input artefacts for inclusion in the knowledge base. The model contains a mechanism to maintain data consistency, and a query engine to respond to trust claim queries invoked from the use cases.

An instantiation of the approach should have a solid basis, particularly for its semantic aspects. Reasoning should be deterministic, decidable and computable in a reasonable time. There should be support for the different use cases and their claims. It should be possible to integrate contextual information in varying formats in a relatively easy way, to include publicly available information. The instantiation of the approach in this article is limited to the TCI model. Instantiating the trust policy validation algorithm and the corresponding trust policies and their assumptions is identified as a further research area. The first requirement indicates the need for a mathematical basis, with a focus on logic. Boolean logic, reputation scoring, subjective logic and description logic were compared, and the latter [1] was selected as the basis for the prototype implementation.

4 A Prototype Implementation

The logical model was defined in the logic \mathcal{SROIQ} and implemented in OWL DL². For a treatment of OWL DL refer to [6, 11]. OWL DL was used because

² <http://www.w3.org/2012/pdf/REC-owl2-direct-semantics-20121211.pdf>.

it is the syntactic fragment of OWL that abides the syntactic restriction that OWL axioms can be read as *SROIQ* axioms for which the structural restrictions are satisfied. This means that once *SROIQ* constructors and axioms are identified, these are described in DL classes and properties. Protégé³ was used as programming environment. A Knowledge Base is a combination of T-, R- and A-boxes. The T- and R-boxes resulted from the modelling. The A-boxes resulted from manual imports. The reasoning capability is provided by the Hermit reasoner [10], built into Protégé. The query engine consists of the DL and SPARQL query interfaces of Protégé. The normative terminology is based on the EU eIDAS and the US FICAM definitions, and the individuals are based on evidence captured from on-line sources. To promote interoperability of the model, existing ontologies were used where possible. The implementation approach is now described.

4.1 Four Step Implementation Approach

Identification of Concepts and Classes. In the first step, *SROIQ* concepts were identified from the eIDAS and FICAM literature and modelled as OWL DL classes. The first concept that emerged was an anchorpoint that oversees supervision and publishes metadata. Supervisors and trust anchors may have a legal basis in a particular jurisdiction, or may be based on less stringent concepts such as a membership agreement. The second set of relations that emerged from this analysis were those between a service provider and consumer, making use of trust services. Such a TSP is overseen by a supervisor. The supervisor can point to the TSP's meta-data from within his own meta-data. This allows services consumers that invoke TSP services to validate against official meta-data. A third set of relations emerged around registers and assurance assessors. Since TSPs provide trust services against remuneration, they are typically officially registered organisations that pay taxes. Assurance assessors review that TSPs meet the requirements imposed on them, and report on this to the relevant supervisory authority. The analysis resulted in eleven concepts, listed in Table 1. The relations between them are not included in this table, but are modelled in the OWL DL model. They implement the description above.

Reuse of Existing Vocabularies. In the second step, as the prototype model aims to be interoperable with existing definitions, vocabularies were evaluated for potential reuse or extension. The W3C list of ontologies⁴ was used as a starting point. The DCMI's *dcterms* vocabulary was found to be the most relevant standard, complemented by the W3C's *Organization*, and *Registered Organization* vocabularies. The first column of Table 1 lists the *SROIQ* concept name. The second column provides a description. The third column indicates the basis for the semantic class. For further refining the class definition, three alternatives are possible. Either an existing vocabulary offers a relevant class that can

³ <http://protege.stanford.edu>.

⁴ http://www.w3.org/wiki/Lists_of_ontologies/.

Table 1. *SRIOQ* concepts and their semantic interpretation

<i>SRIOQ</i> concepts		
<i>SRIOQ</i> conceptname	Description	Semantic implementation
Jurisdiction	The extent or range of judicial, law enforcement, or other authority	Direct use of <i>dcterms:Jurisdiction</i>
LegalBasis	Legislation that provides authority	New subclass of <i>dcterms:BibliographicResource</i>
TrustMetaData-MR	Published meta-data about trust in machine readable format	New subclass of <i>dcterms:BibliographicResource</i>
TrustMetaData-HR	Published meta-data about trust in human readable format	New subclass of <i>dcterms:BibliographicResource</i>
TrustService	Service offering certificates, identity, authentication, time stamping, registered electronic delivery	New subclass of SKOS <i>skos:concept</i>
TrustAnchor	Formal organisation, mandated within some jurisdiction	New subclass of <i>org:FormalOrganization</i>
TrustSupervisor	Formal organisation, mandated within some jurisdiction	New subclass of <i>org:FormalOrganization</i>
TrustServiceProvider	Registered organisation providing trust services	New subclass of <i>regorg:RegisteredOrganization</i>
Register	Organisation that registers and makes available official information about other organisations	New subclass of <i>org:FormalOrganization</i>
TrustServiceAssuranceAssessor	Organisation that assesses the assurance level of a TSP service	New subclass of <i>org:FormalOrganization</i>
ContextualEvidenceProvider	Organisation that provides contextual evidence about an organisation or service	New subclass of <i>org:Organization</i>

directly be used, an existing vocabulary offers a class that can be refined by subclassing it, or no relevant classes from existing vocabularies could be identified. In the latter case, a new class needs to be defined. In the current prototype, this latter alternative was not used. Whatever alternative is used, it yields the T-boxes.

Roles and Properties. In the third step, *SRIOQ* roles were defined and implemented as OWL DL properties. For classes based on existing vocabularies, existing object properties were reused as roles where possible, as well as existing data properties to capture relevant attributes. Otherwise, new definitions were created. This yields the R-boxes.

Individuals. In the fourth step, individuals were created for the different classes of the model, yielding the A-boxes. In the current prototype this has been done manually. However it has been shown [12] that this can be automated using e.g. XSLT transformations.

4.2 Illustration of the Four Steps

The implementation of the *Jurisdiction* class illustrates the direct reuse of existing terminology. It is derived from the eIDAS and FICAM literature there is a need for such a concept, since claims will only be valid within a certain jurisdiction. The DCMI's class *dcterms:jurisdiction* is used directly in the TCI model. Then existing object properties such as *dcterms:coverage* are analysed. To capture the relation between a formal organisation and a jurisdiction, the new object property *hasJurisdiction* is introduced, with domain 'FormalOrganization' and range 'Jurisdiction'. To conclude, two individuals were created, EU Jurisdiction and US Jurisdiction.

The implementation of the *TrustServiceProvider* class illustrates the reuse of existing terminology by subclassing. It is derived there is a need for such a concept, since TSPs are used by both providers and consumers of business services. Both the European and the US regulations define a TSP. The W3C's *org* vocabulary is identified to contain the class *org:FormalOrganization*, and the *regorg* vocabulary contains the class *regorg:RegisteredOrganization*. *TrustAnchor* is subclassed of the latter. Then *isSupervisedOrCertifiedBy* and *providesTS* are created as additional roles. The data property *regorg:legalName* is reused. To conclude, TSP individuals are created.

4.3 Generating Responses to Requests

Once the KB contains T-, R- and A-boxes, and has been classified, queries can be answered. The present prototype implements elements of the validation of claimant functional assumptions for TSPs, TrustSupervisors and TrustAnchors. Generating responses to requests that result from invoking a trust validation policy is specified as illustration. In this case, claimant functional assumptions need to be validated that address the involved TSP and the TrustAnchor. Assumptions on TSP existence can be verified by the DL query '*TrustServiceProvider and registration some and providesTS some*'. This query yields the set of TSP individuals with these properties. Assumptions on TSP meta data and qualifications can be verified by the DL query '*TrustServiceProvider and isSupervisedOrCertifiedBy some and (publishesTMD-HR some or publishesTMD-MR some)*'. Assumptions on the legal basis of a trust supervisor can be verified by '*TrustSupervisor and hasLegalBasis some*'. The response to this query allows the distinction between a trust supervisor operating established on a legal basis (e.g. a national trust supervisor of one of the European countries) and a trust supervisor operating according to a less formal Membership agreement (e.g. the Kantara Initiative).

5 Strengths, Weaknesses and Further Research

Analysing the proposed approach leads to the identification of the following strengths. As the logical model is based on legislation and standards rather

than on technical vocabulary only, it allows an interpretation that spans these two domains. As it builds on existing vocabularies from W3C and Dublin Core it allows interoperability, since rather than reinventing the wheel it starts from a terminology that has been created through large scale consensus. As it has a formal logic basis, composed of *SRIOQ* and OWL DL, it ensure that interpretations conform to the logical definitions. It introduces transparency by allowing invocation of the explanation of the DL classification and inferences. It allows also the inclusion of contextual assertions from sources that are reasonably independent from the actor providing the trust claim.

The current concepts do not address securing the provenance of the various assertions in the knowledge base, as well as their timeliness. Also the formalisation of the degree of independence of providers of contextual assertions from the actors providing the claims is not addressed. The prototype implementation is limited to the TCI model and does not implement the trust policy validation algorithm. The TCI request-response mechanism is currently only simulated by the query interface and does not support an http-like request-response protocol. It is further limited by the fact that individual assertions need to be entered manually.

Areas for further research include instantiating the trust policy validation algorithm, and its deployment in trust policies. Further areas include addressing the weaknesses identified in the preceding section. Securing the provenance and timeliness of the various assertions in the knowledge base, as well as the degree of independence of providers of contextual assertions related to the actors should be more formally addressed. Specialisations towards trust for IdPs and authentication, and towards trust for other TSPs can also be envisaged.

References

1. Baader, F.: *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York (2003)
2. Cofta, P., Trust, C.: *Control: Confidence in a Convergent World*. Wiley, New York (2007)
3. Fuchs, A., Gürgens, S., Rudolph, C.: A formal notion of trust – enabling reasoning about security properties. In: Nishigaki, M., Jøsang, A., Murayama, Y., Marsh, S. (eds.) *IFIPTM 2010. IFIP AICT*, vol. 321, pp. 200–215. Springer, Heidelberg (2010)
4. Gambetta, D.: Can we trust trust? In: Gambetta, D. (ed.) *Trust: Making and Breaking Cooperative Relations*, pp. 213–237. Basil Blackwell, Oxford (1988)
5. Hartig, O.: Querying trust in RDF data with tSPARQL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) *ESWC 2009. LNCS*, vol. 5554, pp. 5–20. Springer, Heidelberg (2009)
6. Hitzler, P., Krötzsch, M., Rudolph, S.: *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC (2009). http://www.semantic-web-book.org/page/Foundations_of_Semantic_Web_Technologies. ISBN: 978-1-4200-9050-5
7. Huang, J., Nicol, D.: *A Calculus of Trust and its Application to PKI and Identity Management*. ACM (2009)

8. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: Proceedings of the 29th Australasian Computer Science Conference, vol. 48, ACSC 2006, pp. 85–94, Darlinghurst, Australia. Australian Computer Society Inc. (2006)
9. Marsh, S.P.: Formalising trust as a computational concept. Ph.D. thesis (1994)
10. Motik, B., Shearer, R., Horrocks, I.: Optimized reasoning in description logics using hypertableaux. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 67–83. Springer, Heidelberg (2007)
11. Rudolph, S.: Foundations of description logics. In: Polleres, A., d’Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P. (eds.) Reasoning Web 2011. LNCS, vol. 6848, pp. 76–136. Springer, Heidelberg (2011)
12. Sel, M.: Using the semantic web to generate trust indicators. In: Paulus, S., Pohlman, N., Reimer, H. (eds.) Securing Business Processes, pp. 106–119. Vieweg+Tuebner, Springer Science +Business Media (2014)
13. Sel, M.: A comparison of trust models. In: Paulus, S., Pohlman, N., Reimer, H. (eds) Securing Business Processes, pp. 206–215. Vieweg+Tuebner, Springer Science+Business Media (2015)
14. Shekarpour, S., Katebi, S.D.: Modeling and evaluation of trust with an extension in semantic web. *Web Semant.* **8**(1), 26–36 (2010)

Trust and Regulation Conceptualisation: The Foundation for User-Defined Cloud Policies

Jörg Kebbedies^(✉), Felix Kluge, Iris Braun, and Alexander Schill

Faculty of Computer Science,
Technische Universität Dresden, 01062 Dresden, Germany
joerg.kebbedies@mailbox.tu-dresden.de

Abstract. In the areas of secrecy or sensitive data management, the public cloud paradigm is not currently well accepted. The root of this problem arises from an inherent structural concept of restricted responsibilities and the lack of trust from the cloud users' perspective.

This work introduces a conceptual approach to user-centric policy management for cloud usage, combined with an underpinning holistic trust approach. Trust has to be established as a separate infrastructural concept determining the level of user adjustability. This approach outlines how provisioning cloud users' policies is combined with agent-based trust establishment. An ontology-driven regulation concept enables formal policy definitions and trustworthy real-time reasoning about current trust levels, policy states, and pending security risks.

Keywords: Trust · Cloud · Regulation · Policy · Ontology · Logic-based semantic

1 Introduction

Although the proliferation of cloud computing seems to gradually be gaining social recognition, the public cloud sector still lacks well-defined user acceptance in specific business areas with high secrecy and privacy requirements. In the past, this issue was discussed in detail in different studies from BITKOM [2] and BSI [4], which have proved that a lack of trust and fear of risk in public cloud services is the main obstacle to common user acceptance.

Trust becomes the fundamental key approach to open the public cloud for use cases with high demands for security and privacy. Once the importance of trust is established, a new problem arises from its nature. The level to which we can be confident that a prescribed security policy controls a given behaviour is the point that defines the level of trust assurance in general. The level at which one can be confident that a behaviour is confined within a prescribed security policy defines the level of trust assurance [3, p. 28] but trust can only be justified through future confidence.

One of the biggest challenges is developing well-suited cloud user control instruments to ascertain the accuracy of ones trust. Following the strategy of

trust enables coupled organisations to gradually stabilise their relationship properties. The new dimensions resulting from trust–confidence in measured properties rather than blind trust–will have an important impact on cloud computing.

In this paper, we propose one instrument to formalise trust.

2 The Principles of Expectation in a Cloud Context

Adapting social principles of expectation to the cloud requires a full understanding of the concepts *Semantic*, *Receipt*, *Success* to develop coordinating and expected pattern of behaviours. The user may define their expectations in terms of policies, role definitions, or regulations for security, privacy, and reliability, but the fact remains that all intended goals are not predictable and this reduces the likelihood that public cloud usage will be accepted in regulated markets.

It is essential to build semantically richer representations of regulations (*Semantic*). An unmistakable interpretation is one of the main factors in achieving reliable technical transformations. The process of defining policies needs formal linguistic instruments to express regulations; these regulations must be independent of specific business domains but should stay readable for people in regard to different legal-requirement categories [1, 8, 17].

The aspect of *Receipt* is strongly involved with trustworthy technical entities. Trustworthy and evaluated cloud entities, acting on behalf of the cloud user, are the foundations of the cloud user’s confidence and extend his policy management scope. Such entities, which are introduced in Sect. 3.1, are technically realised through knowledge-driven cloud agents, able to enforce a cloud user’s policies in a reliable and trustworthy manner [11].

The aspect of *Success* can only be guaranteed through a strong link between the cloud user’s policy definitions and the trustworthy policy-enforcing entities. Only a measurable concept of trust successfully establishes the cloud user’s confidence that his regulation requirements will be reliably enforced and remain compliant in regard to his expectations.

3 The Trust Concept

Following the arguments from Sect. 2, the strong relationship between regulation and trust can be emphasised. Unfortunately, current standardisation efforts like TOSCA [12] to provide flexible and efficient capabilities for service orchestration, service deployment, and cloud-application policy management follow a functionally driven approach; they do not currently provide bases of trust commensurate with their objectives to improve life-cycle processes for cloud-service provisioning and policy management.

Above all other security aspects, the central key concept is a trusted identity. The assurance in identity established by secure authentication is a necessary condition of regulation. Once users securely authenticate an entity based on its claimed identity, a security context has to be established to regulate its states and behaviour.

3.1 Trustworthy Knowledge-Based Agent

The architectural design depicted in Fig. 1 outlines the main process of enforcing different conceptualised policies in a public cloud architecture.

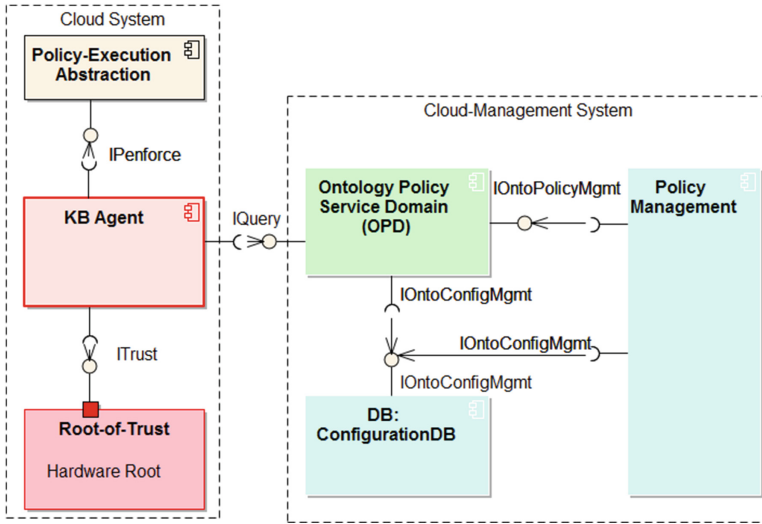


Fig. 1. Knowledge-based agent architecture

The knowledge-based agent (KB Agent) demonstrates that a scope of regulation can only be successfully extended if the transition follows a prepared chain of points of trust. For that reason, each agent has to be technically linked to a root-of-trust (see interface *ITrust*) using the Trust-Establishment-Protocol introduced in Sect. 3.3.

The knowledge-based agent is connected to a knowledge base using the interface *IQuery*, which is represented through a multiple ontology providing formal described regulation instructions. Based on the interface *IPenforce*, the knowledge-based agent is responsible for realising a concept of *Transformation* formally defined in the regulation ontology.

The *Transformation* concept is a bridge between a knowledge-based modelled policy concept and a concrete external cloud system. The subconcepts of *Transformation* are responsible for adapting declarative defined rules into system-specific, technically executable instructions.

Within the prior work [11], different technical approaches were evaluated and some were implemented as part of a proof of concept. The component *Policy-Execution Abstraction* depicted in Fig. 1 represents a Java-based realisation of *Transformation*.

3.2 Trust-Hierarchy Provisioning

Regulations have to find a base of trust as a precondition to effectively acting on the cloud user's intentions. The topic of trust becomes an ingrained part of the concept of regulation, expressing their intrinsic value as a whole.

The model of a Knowledge-based-Agent (KB Agent) approach to controlling policies, depicted in Fig. 1, represents only a small extract of the holistic trust-architectural design concept depicted in Fig. 2, which was introduced in [9] and provides the evaluated base for policy expressiveness and transformation as part of a trust-establishment conceptualisation.

The dynamically established network consists of linked Trust Points, each Trust Point representing a Policy Authority from a regulation point of view. In comparison to social coupling, this kind of architecture claims regions of the cloud user's responsibilities and reflects his dynamically extended scope of regulation. Each established Trust Point acts as a single authority responsible for specific scopes of policy.

The provisioning of Trust Points establishes identifiable entities. The gate to all factors of trust management is the trust in identity [3]. Therefore, the assurance of a secure authentication of identity becomes essential. The process of establishing trustworthy entities has to be combined with the establishment of a cloud-user security context, the user's base of trust on the cloud system's premises. The establishment of a cloud-user security context requires new interfaces for mutual negotiations between user and provider. After a successful negotiation, the cloud user's scope of regulation is extended with the newly established base of trust.

Assuming that each Policy Authority has established a secure session with the central policy knowledge base, the assignment of policies to a specific Policy Authority is declaratively expressed through the method *targetToZone* and is linked to a domain-specific area. The architectural model depicted in Fig. 2 can potentially satisfy different trust-design requirements. The range of specific authorities can be separated, provides a base for modularisation, and enforces principles of separation of duty.

3.3 Trust-Establishment Protocol

The network of trust needs specific policies to regulate the establishment of Trust Points. Besides policies for deployment, actor cooperation, security, and privacy, the current work introduces a specific trust policy to provide a base of linked trusted entities for all further regulation purposes. Such expressivity allows the definition of specific trust policy, negotiating different levels of binding between the cloud user and his trustworthy entities.

The Trust-Establishment-Protocol (TEP) depicted in Fig. 2 is responsible for trust-condition negotiation, starting from a hardware-based root of trust. Once a root of trust is authenticated based on the trust policy, a security context is established through a Trust Point capable of enforcing cloud-specific policies in regard to this regulation layer. Before the next cloud layer can be

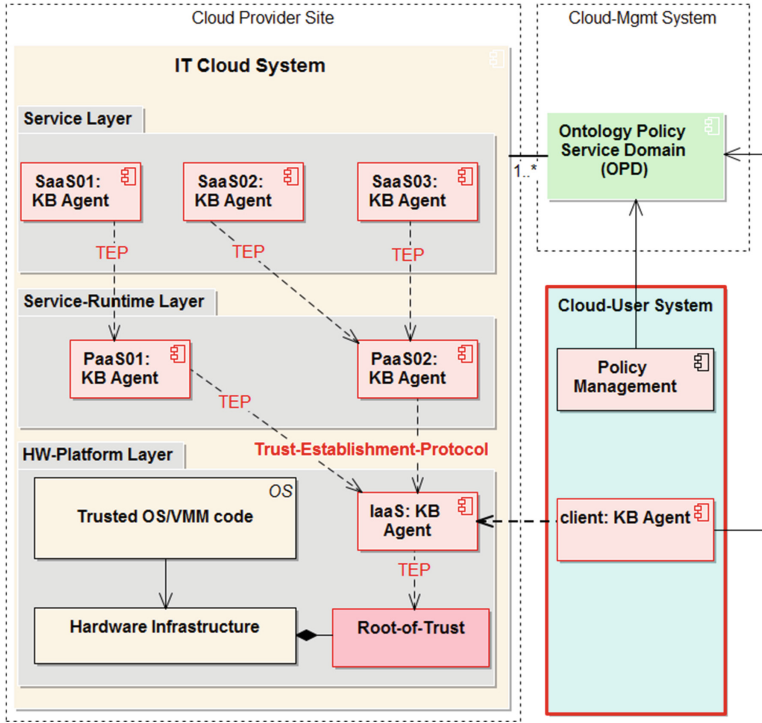


Fig. 2. Trust points: network of policy authorities

regulated, a fundamental security context has to be established and, based on the Trust-Establishment-Protocol, the next trustworthy entity is linked to a chain of trust. The TEP is a cryptographic protocol and uses the TCG Software Stack (TSS) following the TCG version of the TSS specification [16]. The TEP is currently part of the Knowledge-based Agent development.

4 The Ontology Concept

The decision to apply an ontology comes from the demand for a formal representation of knowledge as a base for a precise semantic interpretation of the regulation, domain, and security aspects. Due to its reasoning capability, inferring plays a role for concepts like States, Trust, or Risk, all examples of a represented knowledge that can never be expressed explicitly but is derived from structural or security properties of a target system.

Descending from F-Logic, the ontology language ObjectLogic is used [10]. ObjectLogic extends classical predicate calculus with an object-oriented programming paradigm and follows the closed-world assumption for knowledge representation that assures stable conditions and system states of an expected real

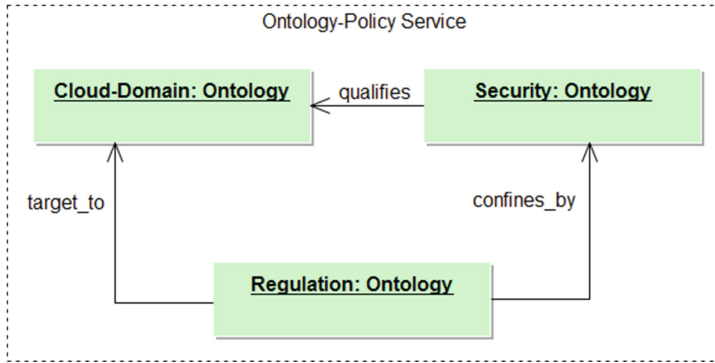


Fig. 3. Multiple-ontology architecture

world. The distributed architecture depicted in Fig. 3 treats the aspect of regulation, the target of regulation, and the aspect of security as separate conceptual frameworks.

Besides the regulation formalisation, the target of regulation, the public cloud, can be formally modelled using an axiomatic language introducing all required cloud concepts as domain vocabulary. From the architectural point of view, the ontology approach allows a system design in stages, starting from some required base concepts that can be formally engineered into more complex system concepts. Both ontologies are well-suited for the demonstration of the base principle in order to establish a structurally and behaviourally regulated concrete cloud system.

The security ontology extends the function-driven domain formalisation with quality-driven concepts like Assets, Confidentiality, and Availability, providing a foundation for the expression of these concepts in authenticated, integrity-protected, or encrypted states, for example [6, 7]. The current work extends the security consideration through a security-model conceptualisation. Security models provide a formal representation of the access-control security policy [13]. The use of a Mandatory Access Control (MAC) model mitigates deficiencies of standard UNIX-based access-control models; the cloud user is given the security background needed to take over responsibility for security management.

The distributed ontology design is still under development; it will be extended based on the evaluation results of the current Cloud-Kit proof-of-concept and will be published in a specialised paper about the conceptualisation approach.

5 The Cloud-Kit Reference Project

The idea of a Cloud Kit is modularisation: the cloud user is faced with a new role as designer of trustworthy cloud services as opposed to his generally accepted service-consuming role, which is influenced by the Trusted Computing Group (TCG) specification standards [15] describing architectural submissions

and processes to establish trusted multi-tenant infrastructures. The main concepts behind the design principles are the Trusted Context and the Trusted System Domain.

The distinction between cloud user and cloud provider remains, but their authorities are fully reviewed and redefined. The cloud user must now select the right conditions for his own architectural design of a cloud foundation commensurate with his compliance requirements. One of the cloud provider's responsibilities is the preparation of well-founded infrastructural environments for the cloud user's independently designed cloud-service concept.

The Trusted Context represents a verified cloud provider's identity and provides cryptographic key artefacts for further mutual negotiations between both parties, thus separating all communication from other cloud users on the same cloud platform. The usage of cryptographic keys for signing and encryption maintains the cloud user's confidence in his connection to the target cloud-provider platform and allows him to adjust the technical preconditions by computing cryptographically signed cloud-platform properties in regard to his base requirements.

The Trusted System Domain is a runtime home base equipped with instruments and controlling resources. Through the use of cryptographic artefacts, it is able to establish a secure channel between the cloud user and the Trusted System Domain.

The cloud architectural reference model was first introduced in [9] and enables the evaluation of a dynamically established interconnected Trust-Point backbone following the model in Sect. 3.2. Rooted in a trusted IT platform layer and reaching the service layer, different Trust Points control ontology-provided policies and trustworthily report the current trust and system state. The proof of concept should resolve the following points:

- **trust policy enforcement:** The proof of concept verifies the roll-out of policy agents based on TEP; they are responsible for policy enforcement and for providing technical interfaces to transform diverse regulation goals.
- **satisfiability of domain concept:** It is important to verify the degree of detail of each object's specification to model an arbitrary cloud architecture.
- **policy coverage:** The policy conceptualisation has to provide a generalisation able to express different governance objectives [5, 14] in order to control specific processing alignments.
- **policy expressiveness:** The concept of constraints largely determines the process of context-oriented regulation refinement. It is important to prove the expressiveness of the underlying constraint conceptualisation in regard to different levels of constraining aspects.
- **policy transformation capabilities:** Transformations induce costs in terms of duration, computing time, and synchronisation, so the question of transformation efficiency remains open.

6 Outlook

The current work demonstrates a fully new approach to cloud system management where trust is deliberately established as a foundation for the cloud user's regulation range, allowing the design of a user-defined cloud service environment.

The issue of the assured system state is currently under development. Once the cloud user can effectively enforce different policies, he needs confidence that the established system state will not change without his knowledge.

During the development of a powerful declarative regulation framework, contractually defined one-way policy control requires extended declarative concepts restricting the cloud provider from influencing running policies defined by the cloud user. Here it is important to integrate the support of different security models into the current regulation conceptualisation.

As part of the cloud-domain ontology, Connections are essential conceptual elements that establish the system state and deploy a horizontally driven relationship model. Each instance of a Connection affects both functional and security policy design.

The concept of Connections has to be extended to introduce the Trust-Establishment-Protocol (TEP) depicted in Fig. 2 and deploy a vertical relationship model. The protocol design is still under development but should become an integrated part of the Connection conceptualisation.

Successfully finalising both the support of extended security models and the regulated establishment of vertical trustworthy Connections provides the foundation for a user-defined cloud policy.

References

1. Androcec, D., Vrcek, N., Seva, J.: Cloud computing ontologies: a systematic review. In: Proceedings of the Third International Conference on Models and Ontology-Based Design of Protocols, Architectures and Services, pp. 9–14 (2012)
2. Barot, P., et al.: Cloud Computing - Evolution in der Technik, Revolution im Business. Ed. by Dr. Mathias Weber. BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (2009)
3. Benantar, M.: Access Control Systems: Security, Identity Management and Trust Models. Springer Science & Business Media, New York (2006)
4. BSI. Sicherheitsempfehlungen für Cloud Computing Anbieter (2012)
5. EUROPEAN COMMISSION. Proposal for a REGULATION 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 (General Data Protection Regulation). Technical report EUROPEAN PARLIAMENT AND OF THE COUNCIL, January 2012. http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf
6. Fenz, S., Ekelhart, A.: Formalizing information security knowledge. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 183–194. ACM (2009)
7. Herzog, A., Shahmehri, N., Duma, C.: An ontology of information security. *Int. J. Inf. Secur. Priv. (IJISP)* 1(4), 1–23 (2007)

8. Humberg, T., Wessel, C., Poggenpohl, D., Wenzel, S., Ruhroth, T., Jürjens, J.: Using ontologies to analyze compliance requirements of cloud-based processes. In: Helfert, M., Desprez, F., Ferguson, D., Leymann, F. (eds.) CLOSER 2013. CCIS, vol. 453, pp. 36–51. Springer, Heidelberg (2014)
9. Kebbedies, J., et al.: Conceptualized policy design for user-regulated trusted clouds. In: UCC 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (2015)
10. Kifer, M., Lausen, G.: F-logic: a higher-order language for reasoning about objects, inheritance, and scheme. In: Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data, SIGMOD 1989, Portland, Oregon, USA, pp. 134–146. ACM (1989). ISBN: 0-89791-317-5. doi:[10.1145/67544.66939](https://doi.org/10.1145/67544.66939). <http://doi.acm.org/10.1145/67544.66939>
11. Kluge, F.: Entwicklung und Konzeption zur Umsetzung einer Transformation von einer ontologischen beschriebenen Policysemantik in eine sichere agentenbasierte Ablaufsteuerung. BSc thesis. Technische Universität Dresden (2016)
12. OASIS. Topology, Orchestration Specification for Cloud Applications Version 1.0. In: Organization for the Advancement of Structured Information Standards, 18 March 2013
13. Ott, A.: Mandatory Rule Set Based Access in Linux: A Multipolicy Security Framework and Role Model Solution for Access Control in Networked Linux Systems. Shaker Verlag GmbH, Aachen (2007). ISBN: 383226423X, 9783832264239
14. Recht, G.: Bundesdatenschutzgesetz (BDSG) (German Edition). CreateSpace Independent Publishing Platform, June 2014. ISBN: 9781500100025. <http://amazon.com/o/ASIN/1500100021/>
15. Trusted Computing Group. TCG TMI Reference Framework (2013)
16. Trusting Computing Group. TCG Software Stack Specification, March 2009. http://www.trustedcomputinggroup.org/resources/tcg_software_stack_tss_specification
17. Youseff, L., Butrico, M., Da Silva, D.: Toward a unified ontology of cloud computing. In: Grid Computing Environments Workshop, GCE 2008, pp. 1–10. IEEE (2008)

A Calculus for Distrust and Mistrust

Giuseppe Primiero^(✉)

Department of Computer Science, Middlesex University, London, UK
G.Primiero@mdx.ac.uk

Abstract. Properties of trust are becoming widely studied in several applications within the computational domain. On the contrary, negative trust attribution is less well-defined and related issues are yet to be approached and resolved. We present a natural deduction calculus for trust protocols and its negative forms, distrust and mistrust. The calculus deals efficiently with forms of trust transitivity and negative trust multiplication and we briefly illustrate some possible applications.

1 Introduction

In various areas of the computational sciences, characterizations of trust are used to identify relevant, secure or preferred sources, channels and contents. For trust interpreted as a first order relation between agents, propagation needs to be considered [3, 5, 10, 11]:

Example 1 (Trust Transitivity). If Alice trusts Bob and Bob trusts Carol; should Alice trust Carol?

This is undesirable in many security contexts. Solutions to this problem include decentralised trust [1], bounded-transitivity in authorization contexts [4], and a constraint by guarantors in [6]. In [18], trust is defined as a second-order property of first-order relations (e.g. of communication) between agents. This is applied in [17] to formulate **SecureND**, a proof-theoretic access control model with an explicit trust function over resources: agents do not trust other agents, but the information they receive from them. Informally, the trust function is defined as follows:

Definition 1 (Trust). *If Alice reads ϕ from Bob and ϕ is consistent with her profile, Alice trusts ϕ and can write it.*

SecureND resolves unintended transitive trust by requiring explicit localisation of trusted messages in the agents' profiles, similar to what suggested in [6].

Recently, research has started considering the different meanings of negative trust [9, 13–15, 20]. In the social sciences distrust is response to lack of information [7, 8] and mistrust is former trust destroyed or healed [19]; the contextual account [13] present mistrust as misplaced trust, untrust as little trust and distrust as no trust. This approach abstracts from the reasons behind the attribution of these evaluations, in favour of a purely quantitative approach. Most of the

remaining contributions do not distinguish mistrust from distrust. Propagation for negative (first-order) trust is formulated as follows [12]:

Example 2 (Untrust Multiplication). If Alice does not trust Bob and Bob does not trust Carol; should Alice trust Carol?

In this paper, we introduce (un)SecureND, an extension of the calculus in [17] with rule-based definitions for negative trust over resources. Here and in the following we use the term *untrust* as neutral with respect to its derivatives *mistrust* for misplacement of trust, and *distrust* for betrayal. Our contribution distinguishes among these two terms, based on the intentional characterization offered in [16]. This calculus also resolves the problem of untrust multiplication. Consider the following modified example:

Example 3 (Intentional Untrust Multiplication). Alice does not trust ϕ from Bob: she believes he sends her *intentionally* false information. Bob does not trust $\neg\phi$ from Carol: he believes she sends him *intentionally* false information. Should Alice trust $\neg\phi$ from Carol?

The question is now better specified and we believe can be answered in the affirmative, given Carol's intention to deceive Bob, and Bob's intention to deceive Alice. The related epistemic action of *distrust* has the following intuitive semantics:

Definition 2 (Distrust). *If Alice reads ϕ from Bob and ϕ is inconsistent with Alice's profile, Alice distrust ϕ and writes $\neg\phi$.*

A distinct case for trust misplacement can be formulated as follows:

Example 4 (Unintentional Untrust Multiplication). Alice reads ϕ from Bob, false in view of her current information: she believes she has *unintentionally* held false information $\neg\phi$. Bob has received ϕ from Carol, who can confirm it to Alice. Should Alice trust ϕ from Carol?

The intuitive semantic meaning of this form of negated trust is as follows:

Definition 3 (Mistrust). *If Alice reads ϕ from Bob, ϕ is inconsistent with Alice's profile and Alice wants to maintain consistency, then she either mistrusts $\neg\phi$; else she refuses ϕ .*

To accept or reject such contradicting information might depend on the number and role of other agents available for confirmation.

The rest of the paper is structured as follows. In Sect. 2 we introduce the natural deduction calculus (un)SecureND: it defines protocols by which agents trust, mistrust or distrust information based on an intentional interpretation of the truth of data transmission; we also briefly cover its meta-theoretical properties. In Sect. 3 we illustrate the restriction to untrust multiplication allowed by this calculus and informally present a possible application to software management, extending the work in [2]. In Sect. 4 we survey further research directions.

2 (un)SecureND

(un)SecureND is a natural deduction calculus defining trust, mistrust and distrust protocols. It formalizes a derivability relation on formulas from sets of assumptions (contexts) as accessibility on resources issued by agents.

Definition 4 (Syntax of (un)SecureND).

$$\begin{aligned}
S^\sim &:= \{A \leq B \leq \dots\} \\
BF^S &:= a^S \mid \phi_1^S \rightarrow \phi_2^S \mid \phi_1^S \wedge \phi_2^S \mid \phi_1^S \vee \phi_2^S \mid \perp \\
mode &:= Read(BF^S) \mid Write(BF^S) \mid Trust(BF^S) \\
RES^S &:= BF^S \mid mode \mid \neg RES^S \\
\Gamma^S &:= \{\phi_1^S, \dots, \phi_n^S\};
\end{aligned}$$

\mathcal{S} is a set of subjects, with a partial order relation \leq over $\mathcal{S} \times \mathcal{S}$: intuitively, $S \leq S'$ means that subject S has higher security privileges than S' . The partial order allows for branching in the hierarchy, so that e.g. $A < B < C$ and $A < B < D$, but C, D are not comparable. BF^S is a set of boolean formulae inductively defined by logical connectives and including \perp for the false. $mode$ is a variable for reading, writing and trusting formulae. Formulae and functions are closed under negation. $\vdash \phi^A$ indicates a validly derivable resource ϕ issued by agent A . Context Γ^A formalises a set of formulae describing the profile for agent A , under which some other resource can be accessed. A context can be extended by a formula issued by the same agent, denoted by Γ^A, ϕ^A ; or it can be extended by resources from a different agent, denoted by Γ^A, ϕ^B and Γ^A, Γ^B .

Definition 5. An (un)SecureND-formula $\Gamma^A \vdash RES^B$ says that under the profile for user A , some resource from user B is validly accessed, given $A \sim B$.

The calculus is based on two sets of rules. The access order to be applied to these rules can be specified dependently on the application: for example, to implement a downwards-only access protocol, the rules will hold only if $A < B$. The operational rules to introduce and eliminate connectives on resources across agents are given in Fig. 1. The rule *Atom* establishes derivability of formulae included in well-formed contexts and preserved under extension. We use the abbreviation wf for a profile that preserves consistency construable by induction from the empty profile. \wedge -I says that if ϕ_1^A is derivable from profile Γ^A and ϕ_2^B is derivable from profile Γ^B , then their conjunction is derivable from the joint profiles. By the elimination, each composing resource is derivable from the combined profiles. \vee -I says that if a joint profile for users A, B can access a formula ϕ_i^I , then it can access the disjunction with any other formula. By the elimination, each resource ψ^I derivable from each component ϕ_i^I can also be obtained by the extended profile. \rightarrow -Introduction establishes the validity of the Deduction Theorem; its elimination implements Modus Ponens. Negation is defined (in the standard constructive way) by implication to the false.

In Fig. 2 we present the access rules allowing a user's profile to act on resources available from another user. \neg -distribution implements a form of negation-completeness: if a profile cannot access a resource from another agent,

$$\begin{array}{c}
\frac{\Gamma^A \vdash wf}{\Gamma^A; \Gamma^B \vdash b} \text{Atom, for any } b \in \Gamma^B \\
\\
\frac{\Gamma^A \vdash \phi_1^A \quad \Gamma^B \vdash \phi_2^B}{\Gamma^A; \Gamma^B \vdash \phi_1^A \wedge \phi_2^B} \wedge\text{-I} \quad \frac{\Gamma^A; \Gamma^B \vdash \phi_1^A \wedge \phi_2^B}{\Gamma^A; \Gamma^B \vdash \phi_i^I} \wedge\text{-E} \\
\\
\frac{\Gamma^A; \Gamma^B \vdash \phi_i^I}{\Gamma^A; \Gamma^B \vdash \phi_1^A \vee \phi_2^B} \vee\text{-I} \quad \frac{\Gamma^A; \Gamma^B \vdash \phi_1^A \vee \phi_2^B \quad \phi_i^I \vdash \psi^I}{\Gamma^A; \Gamma^B \vdash \psi^I} \vee\text{-E}
\end{array}$$

with $I \in \{A, B\}, i \in \{1, 2\}$ in the above rules.

$$\begin{array}{c}
\frac{\Gamma^A; \phi_1^B \vdash \phi_2^B}{\Gamma^A \vdash \phi_1^B \rightarrow \phi_2^B} \rightarrow\text{-I} \quad \frac{\Gamma^A \vdash \phi_1^B \rightarrow \phi_2^B \quad \Gamma^A \vdash \phi_1^B}{\Gamma^A; \phi_1^B \vdash \phi_2^B} \rightarrow\text{-E} \\
\\
\frac{\Gamma^A \vdash RES^A \rightarrow \perp}{\Gamma^A \vdash \neg RES^A} \text{bot}
\end{array}$$

Fig. 1. The system (un)SecureND: operational rules

then it can access its negation (although strong, this rule is essential to preserve consistency). *read* says that from any well-formed profile A , formulae from a profile B can be read (this will hold according to the required constraint on the order relation among agents). *trust* says that if a resource can be read and it preserves consistency when added to the reading profile, then it can be trusted. *write* says that a readable and trustable resource can be written. By DTrust, agent A distrusts a resource ϕ^B if it induces contradiction when read from Γ^A . Its elimination uses \rightarrow -introduction to induce *write* from the receiver profile of any resource that follows distrusting operations. This trivially allows *Write*($\neg\phi^B$) when $\neg\text{Trust}(\phi^B)$ holds. By MTrust, agent A mistrusts resource $\phi^A \in \Gamma^A$ if it contradicts some received ψ^B ; then $Cn(\phi^A)$ is removed to accommodate ψ^B in Γ^A . Its elimination depends on a checking operation. By MTrust-E1, if at least one C agent higher in the order than the sender B verifies the information ϕ^A originally held by the receiver A , ψ^B is rejected; if the receiving agent is the only one higher in the order relation with respect to the sender, the mistrust operation reduces to a distrust one; for $C < B < A$, the receiver A looks for all agents with higher reputation and/or privileges than sender B in order to check for the content of the message ψ . By MTrust-E2, if for every agent C higher than the sender B verifies the received contradictory information ψ^B , the receiver A removes ϕ^A from her profile and trusts the new information.

2.1 Metatheory

The following standard meta-theoretical properties hold for (un)SecureND under trust, all proofs are formulated by structural inductions on the derivation of the second assumption (omitted for brevity).

$$\begin{array}{c}
\frac{\Gamma^A \vdash \neg mode(\phi^B)}{\Gamma^A \vdash mode(\neg\phi^B)} \text{ } \neg\text{-distribution} \qquad \frac{\Gamma^A \vdash wf}{\Gamma^A \vdash Read(\phi^B)} \text{ } read \\
\\
\frac{\Gamma^A \vdash Read(\phi^B) \quad \Gamma^A; \phi^B \vdash wf}{\Gamma^A \vdash Trust(\phi^B)} \text{ } trust \\
\\
\frac{\Gamma^A \vdash Read(\phi^B) \quad \Gamma^A \vdash Trust(\phi^B)}{\Gamma^A \vdash Write(\phi^B)} \text{ } write \\
\\
\frac{\Gamma^A \vdash wf \quad \Gamma^A \vdash Read(\phi^B) \rightarrow \perp}{\Gamma^A \vdash \neg Trust(\phi^B)} \text{ } DTrust\text{-Intro} \\
\\
\frac{\Gamma^A \vdash \neg Trust(\phi^B) \quad \Gamma^A \vdash \neg Trust(\phi^B) \rightarrow \psi^A}{\Gamma^A \vdash Write(\psi^A)} \text{ } DTrust\text{-Elim} \\
\\
\frac{\Gamma^A \vdash Read(\psi^B) \rightarrow \perp \quad \Gamma \setminus \{\phi^A\} \vdash wf, \forall \phi^A \vdash Read(\psi^B) \rightarrow \perp}{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A)} \text{ } MTrust\text{-Intro} \\
\\
\frac{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A) \quad \Delta^C \vdash Read(\psi^B) \rightarrow \perp}{\Gamma^A; \Delta^C \vdash Trust(\phi^A)} \text{ } MTrust\text{-E1, for } C < B \\
\\
\frac{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A) \quad \Delta^C; \psi^B \vdash wf}{\Gamma \setminus \{\phi\}^A; \Delta^C \vdash Trust(\psi^B)} \text{ } MTrust\text{-E2, } \forall C < B
\end{array}$$

Fig. 2. The system (un)SecureND: access rules

Theorem 1 (Weakening $A \sim B$). If $\Gamma^A \vdash Write(\phi^A)$ and $\Gamma^A \vdash Trust(\psi^B)$, then $\Gamma^A; \psi^B \vdash Write(\phi^A)$.

Theorem 2 (Contraction $A \sim B$). If $\Gamma^A, \phi^A; \phi^B \vdash Write(\psi^A)$, then $\Gamma^A, \phi^A \vdash Write(\psi^A)$.

Theorem 3 (Exchange $A \sim B$). If $\Gamma^A, \phi^A; \psi^B \vdash \rho^A$, then $\Gamma^A; \psi^B; \phi^A \vdash \rho^A$.

The general form of the cut rule is as follows:

$$\frac{\Gamma^A \vdash \phi^B \quad \Delta^B, \phi^B \vdash \psi^B}{\Gamma^A; \Delta^B \vdash \psi^B} \text{ } Cut$$

With $A < B$, it amounts to a cut downwards the order relation; with $B < A$ to one upwards: which one is allowed depends again on the application. If $\phi^B \equiv \neg Trust(\phi^B)$ and $A < B$, then the first premise is the result of a DTrust rule, the second premise result from a MTrust rule, and the cut rule eliminates both;

if $\phi^B \equiv \neg \text{Trust}(\phi^A)$, the first premise is obtained by a MTrust rule, the second from a DTrust rule. In all these cases the conclusion of Cut will be an instance of a Weakening rule. If $\psi^B \equiv \neg \text{Trust}(\psi^B)$, then all cases reduce to instances of Weakening on conclusions of a MTrust rule. Then untrust relations safely extend the following from [17]:

Theorem 4 (Cut-Elimination Theorem). *Any (un)SecureND derivation with an instance of a Cut-rule can be transformed into another derivation with the same end sequent iff appropriate trust-access is granted on any upward domination relation among agents.*

3 Examples and Applications

In [17] trust transitivity from Example 1 is resolved by explicitly guaranteeing consistency on every access to resources within the current profile. If Alice trusts ϕ from Bob, and Bob trusts ψ from Carol, Alice also trusts (and eventually writes) information ψ from Carol iff extending her profile Γ^A with information ϕ^B and ψ^C is explicit and preserves consistency.

In (un)SecureND, untrust multiplication from Example 2 is restricted to *dis-trust*, i.e. all agents involved are actively trying to deceive their trustor:

$$\frac{\Gamma^B \vdash wf \quad \Gamma^B \vdash \text{Read}(\neg\phi^C) \rightarrow \perp}{\Gamma^B \vdash \neg \text{Trust}(\neg\phi^C)} \quad \frac{\Gamma^B \vdash \text{Write}(\phi^B) \quad \Delta^A \vdash \text{Read}(\phi^B) \rightarrow \perp}{\Delta^A \vdash \neg \text{Trust}(\phi^B)} \quad \frac{\Delta^A; \neg\phi^C \vdash wf}{\Delta^A \vdash \text{Trust}(\neg\phi^C)} \quad \frac{\Delta^A \vdash \text{Trust}(\neg\phi^C)}{\Delta^A \vdash \text{Write}(\neg\phi^C)}$$

If Alice believes Bob is trying to deceive her with ϕ , and Bob believes Carol is trying to deceive him with $\neg\phi$, then Alice can trust $\neg\phi$ from Carol.

SecureND has been applied to the Minimally Trusted Install Problem in [2]: determine the way to install a new package p in a system such that the minimal amount of transitively trusted dependencies for p is satisfied. In (un)SecureND we can resolve the negative counterpart of this problem. We offer here only an informal explanation and leave a full formalization and the extension of the Coq protocol from [2] to further research. Consider an installation profile Γ^A , and a software package ψ available from repository B for installation. DTrust-Intro can be applied to return all packages that have unresolved conflicts in Γ^A and as such cannot be installed, including ψ^B . DTrust-Elim returns all packages that can be installed under the current conflict with ψ^B . MTrust-Intro returns all packages already installed in Γ^A that need to be removed for Γ^A to install ψ^B safely. MTrust-E1 returns all external packages that can be installed in Γ^A preserving the *current* installation and hence the conflict with ψ^B . MTrust-E2 returns all packages that can be safely installed in Γ^A preserving the installation of ψ^B .

4 Conclusions

(Un)trust relations reveal relevant problems for privacy and security. Attackers can exploit negative trust to induce unconstrained positive information; intentional transmission of true data can be conceived as a strategy to win the trustor's confidence for future attacks, with trustworthiness evaluation based on records of high rate of false alarms (or low records of true alarms). Untrust multiplication can generate unintended accesses and operations. An evaluation based on intentionality criteria can offer a sensibly better solution in many cases if combined with a quantitative and computationally feasible approach. We have presented a calculus for access control protocols with negative trust, modelled formally as functions on resources issued by agents. This language qualifies trust transitivity under consistency constraints and limits untrust multiplication to intentional cases of false data transmission. It also allows revision of false content held within an agent's profile in the form of mistrust. Next stages of this research will focus on defining structural weakenings of the calculus and the development of applications.

References

1. Abdul-Rahman, A., Hailes, S.: A distributed trust model. In: Haigh, T., Blakley, B., Zurko, M.E., Meodaws, C. (eds.), *Proceedings of the 1997 Workshop on New Security Paradigms*, Langdale, Cumbria, United Kingdom, September 23–26, 1997, pp. 48–60. ACM (1997)
2. Boender, J., Primiero, G., Raimondi, F.: Minimizing transitive trust threats in software management systems. In: Ghorbani, A.A., Torra, V., Hisil, H., Miri, A., Koltuksuz, A., Zhang, J., Sensoy, M., García-Alfaro, J., Zincir, I. (eds.) *13th Annual Conference on Privacy, Security and Trust, PST 2015*, Izmir, Turkey, July 21–23, 2015, pp. 191–198. IEEE (2015)
3. Chakraborty, P.S., Karform, S.: Algorithms, designing trust propagation based on simple multiplicative strategy for social networks. *Procedia Technol.* **6**, 534–539 (2012). 2nd International Conference on Communication, Computing & Security [ICCCS-2012]
4. Chapin, P.C., Skalka, C., Wang, X.S.: Authorization in trust management: features and foundations. *ACM Comput. Surv.* **40**(3), 1–48 (2008)
5. Christianson, B., Harbison, W.S.: Why isn't trust transitive? In: Crispo, B. (ed.) *Security Protocols 1996*. LNCS, vol. 1189, pp. 171–176. Springer, Heidelberg (1997)
6. Christianson, B.: Trust*: using local guarantees to extend the reach of trust. In: Christianson, B., Malcolm, J.A., Matyáš, V., Roe, M. (eds.) *Security Protocols 2009*. LNCS, vol. 7028, pp. 179–188. Springer, Heidelberg (2013)
7. Cvetkovich, G.: The attribution of social trust. In: Cvetkovich, G., Lofstedt, R. (eds.) *Social Trust and the Management of Risk*, pp. 53–61. Earthscan, London (1999)
8. Cvetkovich, G., Lofstedt, R.E.: Social trust and culture in risk management. In: Cvetkovich, G., Lofstedt, R. (eds.) *Social Trust and the Management of Risk*, pp. 9–21. Earthscan, London (1999)

9. Guha, R.V., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, New York, NY, USA, May 17–20, 2004, pp. 403–412 (2004)
10. Jamali, M., Ester, M.: A Matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 135–142. ACM, New York (2010)
11. Jøsang, A., Marsh, S., Pope, S.: Exploring different types of trust propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)
12. Jøsang, A., Pope, S.: Semantic constraints for trust transitivity. In: Hartmann, S., Stumptner, M. (eds.), APCCM, vol. 43 of CRPIT, pp. 59–68. Australian Computer Society (2005)
13. Marsh, S., Dibben, M.R.: Trust, untrust, distrust and mistrust – an exploration of the Dark(er) side. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 17–33. Springer, Heidelberg (2005)
14. McKnight, D.H., Chervany, N.L.: Trust and distrust definitions: one bite at a time. In: Falcone, R., Singh, M., Tan, Y.-H. (eds.) AA-WS 2000. LNCS (LNAI), vol. 2246, pp. 27–54. Springer, Heidelberg (2001)
15. McKnight, D.H., Kacmar, C., Choudhury, V.: Whoops..did i use the wrong concept to predict e-commerce trust? Modeling the risk-related effects of trust versus distrust concepts. In: 36th Hawaii International Conference on System Sciences (HICSS-36 2003), CD-ROM / Abstracts Proceedings, January 6–9, 2003, Big Island, HI, USA, p. 182 (2003)
16. Primiero, G., Kosolovsky, L.: The semantics of untrustworthiness. *Topoi* **35**(1), 253–266 (2013)
17. Primiero, G., Raimondi, F.: A typed natural deduction calculus to reason about secure trust. In: Miri, A., Hengartner, U., Huang, Audun Jøsang, N.-F., García-Alfaro, J. (eds.), 2014 Twelfth Annual International Conference on Privacy, Security and Trust, Toronto, ON, Canada, July 23–24, 2014, pp. 379–382. IEEE (2014)
18. Primiero, G., Taddeo, M.: A modal type theory for formalizing trusted communications. *J. Appl. Logic* **10**, 92–114 (2012)
19. Sztompka, P.: *Trust: A Sociological Theory*. Cambridge University Press, Cambridge (1999)
20. Ziegler, C.-N., Lausen, G.: Propagation models for trust and distrust in social networks. *Inf. Syst. Front.* **7**(4–5), 337–358 (2005)

Author Index

- Angulo, Julio 3
- Bandyszak, Torsten 96
- Barni, Gabriele 96
- Basu, Anirban 52
- Behrooz, Saghar 79
- Braun, Iris 174
- Dibben, Mark 137
- Dwyer, Natasha 137
- Fischer-Hübner, Simone 3
- Gil Pérez, Manuel 129
- Giotis, Giorgos 96
- Gol Mohammadi, Nazila 33
- Goldsteen, Abigail 96
- Gómez Mármol, Félix 129
- Hartenstein, Sandro 96
- Heisel, Maritta 33
- Jiang, Lijun 146
- Johansen, Christian 154
- Jøsang, Audun 154
- Kalogiros, Costas 96
- Karegar, Farzaneh 3
- Kebbedies, Jörg 174
- Kiyomoto, Shinsaku 52
- Kluge, Felix 174
- Kwok, Lam-For 146
- Li, Wenjuan 146
- Liu, Yang 17, 113
- Marsh, Stephen 52, 79, 137
- Martínez Pérez, Gregorio 129
- Melas, Panos 96
- Meng, Weizhi 146
- Moffie, Micha 96
- Muller, Tim 17, 113
- Nasser, Bassem I. 96
- Pedersen, Tore 154
- Primiero, Giuseppe 183
- Pulls, Tobias 3
- Rahman, Mohammad Shahriar 52
- Schill, Alexander 174
- Sel, Marc 164
- Serna, Jetzabel 63
- Veseli, Fatbardh 63
- Wang, Dongxia 17
- Weyer, Thorsten 96
- Zhang, Jie 17, 113