Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
• seminars
• proceedings
• dissertations
• thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-675-6

This volume contains the contributions to the 9th conference of the GI special interest group "Sicherheit, Schutz und Zuverlässigkeit" that took place in Konstanz on April 25-27, 2018. The main aspects of the conference were privacy, secure software development, critical infrastructures, security policies, digital forensics, authentication, usability, and cloud computing. Bringing together experts with scientific and practical experience in security and safety is one of the goals of Sicherheit 2018.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018

281

GI-Edition

Lecture Notes in Informatics

Hanno Langweg, Michael Meier, Bernhard C. Witt, Delphine Reinhardt (Hrsg.)

Sicherheit 2018

Sicherheit, Schutz und Zuverlässigkeit

Beiträge der 9. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

25.–27. April 2018
Konstanz

Proceedings

Hanno Langweg, Michael Meier,
Bernhard C. Witt, Delphine Reinhardt (Hrsg.)

SICHERHEIT 2018

Sicherheit, Schutz und Zuverlässigkeit

Konferenzband der 9. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

25.-27. April 2018
in Konstanz

Gesellschaft für Informatik e.V. (GI)

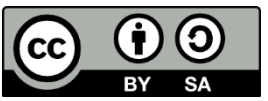
Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-281

ISBN 978-3-88579-675-6
ISSN 1617-5468

Volume Editors
Prof. Dr. Hanno Langweg
HTWG Konstanz, Alfred-Wachtel-Str. 8, 78462 Konstanz, hanno.langweg@htwg-konstanz.de
Prof. Dr. Michael Meier
Univ. Bonn und Fraunhofer FKIE, Endenicher Allee 19a, 53115 Bonn, mm@cs.uni-bonn.de
Bernhard C. Witt
it.sec GmbH & Co. KG, Einsteinstr. 55, 89077 Ulm, bcwitt@it-sec.de
Prof. Dr. Delphine Reinhardt
Universität Göttingen, Goldschmidtstr. 7, 37077 Göttingen, reinhardt@cs.uni-goettingen.de

printed by Köllen Druck+Verlag GmbH, Bonn



Vorwort

Die SICHERHEIT 2018 ist die neunte Ausgabe der regelmäßig stattfindenden Fachtagung des Fachbereichs „Sicherheit – Schutz und Zuverlässigkeit“ der Gesellschaft für Informatik e.V. Sie bietet einem Publikum aus Forschung, Entwicklung und Anwendung ein Forum zur Diskussion von Herausforderungen, Trends, Techniken und neuesten wissenschaftlichen und industriellen Ergebnissen. Die Tagung deckt alle Aspekte der Sicherheit informationstechnischer Systeme ab und versucht eine Brücke zu bilden zwischen den Themen IT Security, Safety und Dependability.

Die SICHERHEIT ist mit fünfzehn Jahren vergleichsweise jung und hat doch eine lange Tradition, die mit der Vorläufertagung VIS Verlässliche Informationssysteme bis Anfang der 1990er Jahre zurückreicht. Auch Konstanz hat mit dem Konstanzer Konzil (1414-1418) eine lange Tradition. Auf der SICHERHEIT wird zwar kein Papst gewählt, sie gehört aber zu den wichtigsten Foren der deutschsprachigen Sicherheits-Community.

Auf der SICHERHEIT lässt sich leicht ein Überblick gewinnen über die Forschungsthemen, die für Security und Safety von IT-Systemen aktuell sind. Aus 53 Einreichungen haben wir 26 Beiträge für das Vortragsprogramm der Tagung ausgewählt: 18 wissenschaftliche Beiträge, 3 im Practitioners Track und 5 im Doktorandenforum.

Die Vielfalt der Beiträge und Teilnehmenden erlaubt es, Kontakte zu knüpfen und zu pflegen und seinen eigenen Horizont zu erweitern. Jungen Wissenschaftlerinnen und Wissenschaftlern bieten wir im Doktorandenforum Gelegenheit zum Austausch. Für die Keynotes konnten wir Prof. Dr. Marc Strittmatter (HTWG Konstanz), Marina Krotofil (FireEye) und Dirk Fox (Secorvo) gewinnen.

Zum ersten Mal findet die SICHERHEIT in Baden-Württemberg beinahe in der Schweiz statt und zum ersten Mal ist der Tagungsort eine Hochschule für angewandte Wissenschaften. Forschung in Safety und Security ist heterogener geworden und verankert an Universitäten, Hochschulen für angewandte Wissenschaften, Forschungsinstituten und in der Wirtschaft. Diese Vielfalt spiegelt auch das große Programmkomitee wider. Mehr als 200 ehrenamtliche Mitglieder der deutschsprachigen Community haben ihren Sachverstand beigesteuert und leidenschaftlich diskutiert.

Unser Dank gilt allen, die zum Gelingen der Tagung beigetragen haben, im Großen wie im Kleinen, im Lichte wie im Schatten. Viele Menschen haben sich engagiert im Verfassen von Beiträgen, in der Begutachtung, in der Auswahldiskussion, beim Zusammenstellen dieses Tagungsbands, im Vortrag und Gespräch während der Tagung, und sichtbar und unsichtbar hinter den Kulissen. Ihnen allen herzlichen Dank.

Hanno Langweg
Michael Meier
Jürgen Neuschwander
Delphine Reinhardt
Bernhard C. Witt

Konstanz, im April 2018

Sponsoren

Wir danken den folgenden Unternehmen für die Unterstützung der Konferenz.

Platin-Sponsor
DB Systel GmbH

Gold-Sponsor
Siemens Postal, Parcel & Airport Logistics GmbH

Silber-Sponsor
ERNW Research GmbH

Silber-Sponsor
genua gmbh

Fachbereich Sicherheit – Schutz und Zuverlässigkeit

Der GI-Fachbereich "Sicherheit - Schutz und Zuverlässigkeit" wurde 2002 gegründet und vernetzt zwei „Communities“ miteinander: Während die „Safety-Community“ vor allem den Schutz der Umwelt vor IT-Systemen (beispielsweise Sicherheit des Menschen vor schwerwiegenden Systemfehlern in Flugzeugen, Kernreaktoren und Kraftwerken) sowie Fehlertoleranzmaßnahmen (z.B. Systemausfälle als Folge von Ermüdungserscheinungen, Softwarefehlern und Naturereignissen) im Blick hat, beschäftigt sich die „Security-Community“ hauptsächlich mit dem Schutz der IT-Systeme und ihrer Umgebung vor Bedrohungen von außen, insbesondere vor Gefahren, die von bösartigen Angriffen (durch Menschen) ausgehen.
Sicherheit ist ein Querschnittsthema. Für den Fachbereich gilt daher eine hohe Flexibilität hinsichtlich der Möglichkeiten zur Quervernetzung verschiedener Gruppen und Themen innerhalb und außerhalb der GI. Diese Quervernetzung wird sowohl in gemeinsamen Veranstaltungen als auch in der starken Berücksichtigung anderer Themen aus der Informatik deutlich. Sicherheit ist kein Selbstzweck, sondern wichtig zur Erfüllung gesellschaftlicher und wirtschaftlicher Bedürfnisse.

Tagungsleitung der Sicherheit 2018

Hanno Langweg, HTWG Konstanz (General Chair)
Michael Meier, Universität Bonn und Fraunhofer FKIE (Program Co-Chair)
Jürgen Neuschwander, HTWG Konstanz
Bernhard C. Witt, it.sec GmbH & Co. KG

Satz des Tagungsbands

Felix Schuckert, HTWG Konstanz

Programmkomitee

Chair: Hanno Langweg, HTWG Konstanz
Program Co-Chair: Michael Meier, Universität Bonn und Fraunhofer FKIE

Herbert Leitold	A-SIT
Emmanuel Benoist	Berner Fachhochschule
Eric Dubuis	Berner Fachhochschule
Isabel Münch	BSI
Steffen Helke	BTU Cottbus-Senftenberg
Sebastian Schmerl	Computacenter
Christian Dietrich	CrowdStrike
Dieter Hutter	DFKI
Arjen Lenstra	EPFL Lausanne
Bernhard Fechner	Fernuniversität Hagen
Jörg Keller	Fernuniversität Hagen
Marko Schuba	FH Aachen
Matthias Hudler	FH Campus Wien
Eckehard Hermann	FH Hagenberg
Markus Zeilinger	FH Hagenberg
Robert Kolmhofer	FH Hagenberg
Ingrid Schaumüller-Bichl	FH Hagenberg
Wilhelm Zugaj	FH Joanneum
Nils Gruschka	FH Kiel
Sebastian Schinzel	FH Münster
Dominik Engel	FH Salzburg
Johann Haag	FH St. Pölten
Simon Tjoa	FH St. Pölten
Paul Tavolato	FH St. Pölten
Annika Meyer	FH Südwestfalen
Steffen Wendzel	FH Worms und Fraunhofer FKIE
Martin Kappes	Frankfurt University of Applied Sciences
Heiko Roßnagel	Fraunhofer IAO
Christoph Krauß	Fraunhofer SIT
Michael Waidner	Fraunhofer SIT
Matthias Wählisch	FU Berlin
Jörn Eichler	FU Berlin
Christoph Meinel	Hasso Plattner Institut

Klaus-Peter Kossakowski	HAW Hamburg
Martin Hübner	HAW Hamburg
Thomas Schmidt	HAW Hamburg
Bettina Buth	HAW Hamburg
Johann Uhrmann	HAW Landshut
Martin Hobelsberger	HAW München
Klaus Junker-Schilling	HAW Würzburg-Schweinfurt
Kristin Weber	HAW Würzburg-Schweinfurt
Erik Buchmann	HFT Leipzig
Holger Morgenstern	Hochschule Albstadt-Sigmaringen
Martin Rieger	Hochschule Albstadt-Sigmaringen
Tobias Heer	Hochschule Albstadt-Sigmaringen
Dominik Merli	Hochschule Augsburg
Kerstin Lemke-Rust	Hochschule Bonn-Rhein-Sieg
Evren Eren	Hochschule Bremen
Andreas Heinemann	Hochschule Darmstadt
Harald Baier	Hochschule Darmstadt
Jens-Peter Akelbein	Hochschule Darmstadt
Marian Margraf	Hochschule Darmstadt
Michael Braun	Hochschule Darmstadt
Martin Stiemerling	Hochschule Darmstadt
Christoph Busch	Hochschule Darmstadt
Dirk Rabe	Hochschule Emden-Leer
Patrick Felke	Hochschule Emden-Leer
Carsten Link	Hochschule Emden-Leer
Wolfgang Ehrenberger	Hochschule Fulda
Peter Heinzmann	Hochschule für Technik Rapperswil
Christoph Reich	Hochschule Furtwangen
Dirk Koschützki	Hochschule Furtwangen
Friedbert Kaspar	Hochschule Furtwangen
Hermann Strack	Hochschule Harz
Andreas Mayer	Hochschule Heilbronn
Manuel Duque-Anton	Hochschule Kaiserslautern
Ingo Stengel	Hochschule Karlsruhe
Peter Dencker	Hochschule Karlsruhe
Norbert Schultes	Hochschule Koblenz
Bernhard Hämmerli	Hochschule Luzern
Konrad Marfurt	Hochschule Luzern
Sachar Paulus	Hochschule Mannheim
Christian Hummert	Hochschule Mittweida
Philipp Brune	Hochschule Neu-Ulm
Christian Bachmeir	Hochschule Neu-Ulm
Dirk Westhoff	Hochschule Offenburg
Erik Zenner	Hochschule Offenburg
Stephan Trahasch	Hochschule Offenburg
Alfred Scheerhorn	Hochschule Osnabrück
Marcus Schöller	Hochschule Reutlingen

Ulrich Greveler	Hochschule Rhein-Waal
Andreas Noack	Hochschule Stralsund
Martin Staemmler	Hochschule Stralsund
Konstantin Knorr	Hochschule Trier
Christoph Karg	HTW Aalen
Jürgen Neuschwander	HTWG Konstanz
Björn Scheuermann	HU Berlin
Johannes Köbler	HU Berlin
Ioannis Krontiris	Huawei
Stefan Dietzel	Humboldt-Universität zu Berlin
Matthias Schunter	Intel
Bernhard C. Witt	it.sec GmbH & Co. KG
Peer Reymann	ITQS GmbH
Jürgen Schönwälder	Jacobs University Bremen
Lothar Fritsch	Karlstad University
Melanie Volkamer	Karlstad University
Andy Rupp	KIT
Hubert Keller	KIT
Willi Geiselmann	KIT
Nils gentschen Felde	LMU München
Florian Alt	LMU München
Basel Katt	NTNU Norwegian Univ. of Science and Technology
Laura Georg	NTNU Norwegian Univ. of Science and Technology
Andreas Aßmuth	Ostbayerische TH Amberg-Weiden
Christoph Skornia	Ostbayerische TH Regensburg
Ina Schiering	Ostfalia HAW
Peter Schwabe	Radboud University Nijmegen
André Egners	Rohde & Schwarz Cybersecurity GmbH
Amir Moradi	Ruhr-Universität Bochum
Markus Dürmuth	Ruhr-Universität Bochum
Thorsten Holz	Ruhr-Universität Bochum
Tim Güneysu	Ruhr-Universität Bochum
Georg Neugebauer	RWTH Aachen
Ulrike Meyer	RWTH Aachen
Karl-Erwin Grosspietsch	Sankt Augustin
Volkmar Lotz	SAP
Sebastian Schrittwieser	SBA Research
Jens Braband	Siemens und TU Braunschweig
Klaus-Michael Koch	TECHNIKON Forschungsgesellschaft mbH
Claus Vielhauer	TH Brandenburg
Andreas Berl	TH Deggendorf
Martin Schramm	TH Deggendorf
Hans-Joachim Hof	TH Ingolstadt
Luigi Lo Iacono	TH Köln
Stefan Karsch	TH Köln
Hans Ludwig Stahl	TH Köln
Ramin Tavakoli Kolagari	TH Nürnberg

Florian Tschorsch	TU Berlin
Admela Jukan	TU Braunschweig
Ansgar Kellner	TU Braunschweig
Wael Adi	TU Braunschweig
Alfred Nordmann	TU Darmstadt
Heiko Mantel	TU Darmstadt
Kay Hamacher	TU Darmstadt
Matthias Hollick	TU Darmstadt
Mira Mezini	TU Darmstadt
Neeraj Suri	TU Darmstadt
Thorsten Strufe	TU Dresden
Peter Lipp	TU Graz
Raphael Spreitzer	TU Graz
Chris Brzuska	TU Hamburg-Harburg
Dieter Gollmann	TU Hamburg-Harburg
Georg Carle	TU München
Michael Franz	UC Irvine
Marit Hansen	Unabh. Landeszentr. f. Datenschutz Schleswig-Holst.
Sven Kuhlmann	Uni Magdeburg
Doğan Kesdoğan	Uni Regensburg
Dominik Herrmann	Universität Bamberg
Emanuel von Zezschwitz	Universität Bonn
Jernej Tonejc	Universität Bonn
Michael Nüsken	Universität Bonn
Peter Martini	Universität Bonn
Rolf Drechsler	Universität Bremen
Thomas Kemmerich	Universität Bremen
Maritta Heisel	Universität Duisburg-Essen
Bernardo Magri	Universität Erlangen-Nürnberg
Felix Freiling	Universität Erlangen-Nürnberg
Francesca Saglietti	Universität Erlangen-Nürnberg
Kai Rannenberg	Universität Frankfurt
Sebastian Pape	Universität Frankfurt
Dieter Hogrefe	Universität Göttingen
Delphine Reinhardt	Universität Göttingen
Lena Wiese	Universität Göttingen
Hannes Federrath	Universität Hamburg
Mathias Fischer	Universität Hamburg
Sascha Fahl	Universität Hannover
Rainer Böhme	Universität Innsbruck
Ruth Breu	Universität Innsbruck
Eberhard Zehendner	Universität Jena
Arno Wacker	Universität Kassel
Henning Schnoor	Universität Kiel
Peter Schartner	Universität Klagenfurt
Daniel Strüber	Universität Koblenz-Landau
Jan Jürjens	Universität Koblenz-Landau

Katharina Bräunlich	Universität Koblenz-Landau
Rüdiger Grimm	Universität Koblenz-Landau
Viorica Sofronie-Stokkermans	Universität Koblenz-Landau
Volker Riediger	Universität Koblenz-Landau
Michael Sonntag	Universität Linz
Maciej Liskiewicz	Universität Lübeck
Rüdiger Reischuk	Universität Lübeck
Christian Kraetzer	Universität Magdeburg
Frank Ortmeier	Universität Magdeburg
Jana Dittmann	Universität Magdeburg
Frederik Armknecht	Universität Mannheim
Bernd Freisleben	Universität Marburg
Nils Aschenbruck	Universität Osnabrück
Eric Bodden	Universität Paderborn
Saqib A. Kakvi	Universität Paderborn
Tibor Jager	Universität Paderborn
Hans P. Reiser	Universität Passau
Hermann De Meer	Universität Passau
Joachim Posegga	Universität Passau
Rolf Schillinger	Universität Regensburg
Van Bang Le	Universität Rostock
Yang Zhang	Universität des Saarlandes
Volkmar Pipek	Universität Siegen
Erhard Plödereder	Universität Stuttgart
Ralf Kuesters	Universität Stuttgart
Christoph Bösch	Universität Ulm
Frank Kargl	Universität Ulm
Burkhard Stiller	Universität Zürich
Andreas Peter	University of Twente
Christian Rohner	Uppsala University
Norbert Pohlmann	Westfälische Hochschule
Mark Strembeck	Wirtschaftsuniversität Wien
Marc Rennhard	ZHAW
Stephan Neuhaus	ZHAW

Practitioners Track

Chair: Bernhard C. Witt, it.sec GmbH & Co. KG

Claus Stark	CitiGroup
Ivan Buetler	Compass Security
Frank Damm	Deutsche Bahn
Manuela Reiss	dokuit
Arslan Brömme	FG BIOSIG
Harald Vater	Giesecke & Devrient
Dirk Koschützki	Hochschule Furtwangen
Jürgen Neuschwander	HTWG Konstanz

Marcel Winandy	Huawei
Jens Nedon	IABG
Karin Schuler	Karin Schuler - Datenschutz und Datensicherheit
Bastian Braun	mgm security partners GmbH
Patrizia Russ	Munich Re
Klaus Kirst	Präsidium für Technik, Logistik u. Verwaltung, Hessen
Torsten Schütze	Rohde & Schwarz Cybersecurity GmbH
Lukas Rist	The Honeynet Project
Hans Pongratz	TU München
Jörn Voßbein	UIMC
Adrian Spalka	Universität Bonn
Michael Meier	Universität Bonn und Fraunhofer FKIE
Michael Heinl	Universität Ulm
Andreas Schaad	Wibu-Systems

Doktorandenforum

Chair: Delphine Reinhardt, Universität Göttingen

Christoph Striecks	AIT Austria
Kerstin Lemke-Rust	Hochschule Bonn-Rhein-Sieg
Andreas Heinemann	Hochschule Darmstadt
Anja Lehmann	IBM Research, Zürich
Ina Schiering	Ostfalia HAW
Jörg Schwenk	Ruhr-Universität Bochum
Luigi Lo Iacono	TH Köln
Dieter Gollmann	TU Hamburg-Harburg
Michael Meier	Universität Bonn und Fraunhofer FKIE
Felix Freiling	Universität Erlangen-Nürnberg
Kai Rannenberg	Universität Frankfurt
Hannes Federrath	Universität Hamburg
Joachim Posegga	Universität Passau

Zusätzliche Gutachter/innen

Maximilian Blochberger	Manuel Koschuch	Marcin Robak
Michael Brunner	Benjamin Krumnow	Mirja Rötting
Christoph Döpmann	Sebastian Kurowski	Christopher Schmitz
Frank Fuhlbrück	Benjamin Leiding	Kenneth Schmitz
Katharina Großer	Marcel von Maltitz	Louis Tajan
Tobias Hamann	Karola Marky	Sree Harsha Totakura
David Harborth	Sadaf Momeni	Cuong Tran
Majid Hatamian	Johannes Mueller	Ingrid Verbauwhede
Maximilian Hb	David Niehues	Ahmed Seid Yesuf
Vladimir Herdt	Christoph Piechula	Ephraim Zimmer
Alina Khayretdinova	Daniel Rausch	

Inhaltsverzeichnis

Privacy

Software

Kritische Infrastrukturen, Policies und Digitale Forensik

Authentisierung und eVoting

Cloud

Practitioners Track

Doktorandenforum Sicherheit 2018

Autorenverzeichnis

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 17

Towards an Architecture for Pseudonymous E-Commerce

Applying Privacy by Design to Online Shopping

Sebastian Pape[1], Daniel Tasche[2], Iulia Bastys[1,3], Akos Grosz[1], Jörg Lässig[2], Kai Rannenberg[1]

Abstract: In this paper we apply privacy by design in e-commerce. We outline the requirements of a privacy-aware online shopping platform that satisfies the principle of data minimization and we suggest several architectures for building such a platform. We then compare them according to four dimensions: privacy threats, transparency, usability and compatibility with existing business models. Based on the comparison, we aim to build the selected platform in the next step.

Keywords: privacy by design; pseudonymity; data minimisation; online shopping; e-commerce

1 Introduction

E-commerce is playing an increasingly important role for operators of shopping platforms and their customers. The estimated revenue for the German e-commerce market in 2017 amounts to €55 billion, while recent statistics forecast 58 million users and a market volume of €78 billion in 2021 [St16]. Shopping platform operators are collecting customer data, as personalized offers and recommendations lead to higher revenues. Despite an increase in public's awareness on the issue of data protection and growing concerns about the usage of their data, currently e-commerce users have no alternative to disclosing personal data and revealing shopping behavior [Jo16]. A recent study reveals that 50% of online services send full information about the users' baskets to Paypal (if PayPal was selected as payment method), which in turn forwards the information, *who* purchased *what* and *where*, to a third-party specialized in data aggregation [Pr16]. At least in Europe, these issues begin to be addressed through several regulations and directives. The General Data Protection Regulation (GDPR), planned to be applied in May 2018 in the EU countries, requires data protection by design and by default: "The controller shall implement appropriate technical and organisational measures, such as *pseudonymisation*, which are designed to implement data-protection principles, such as data *minimisation* [. . .] in order to [. . .] protect the rights of data subjects" [Re16]. Therefore, we aim to improve the processes in e-commerce in respect to the data protection principles pseudonymisation and data minimisation.
The e-shopping platform could track the user's online activity through IP address, third party web-tracking [MM12], browser fingerprinting [Ec10], canvas fingerprinting [Ac14],

[1] Goethe University, Chair of Mobile Business & Multilateral Security , firstname.lastname@m-chair.de
[2] University of Applied Sciences Zittau/Görlitz, {d.tasche,j.laessig}@hszg.de
[3] Chalmers University of Technology, Gothenburg, Swedenbastys@chalmers.se

doi:10.18420/sicherheit2018_01

or evercookies [Ac14]. We do not investigate them in this paper, as previous work has already suggested different countermeasures [DMS04, PCM13, Ba13, LRB16].
Following these requirements, we make a step forward in preserving customer's privacy in online shopping, by describing an e-commerce platform that satisfies the principles of data minimization and pseudonymization (Section 3). We then suggest several architectures for building such a platform (Section 4) and compare them based on the privacy threat analysis methodology LINDDUN [WJ15], but also with respect to usability, transparency and compatibility to existing business models (Section 5).

2 Related Work

Growing concern about user traceability when making electronic payments propelled efforts in the area of privacy-preserving e-commerce. Initial work mainly concentrates on anonymous electronic payment methods through cryptographic mechanisms such as blind signatures [Ch83, Ch85, CFN90]. Aiello et al. [AIR01] describe a cryptographic protocol for anonymous shopping of digital goods based on priced oblivious-transfer and private information retrieval [Ch95]. In their setting, the customer makes an initial deposit which is later used to retrieve the desired items. Besides the initial deposit and the interaction with the platform, the online shop learns nothing else. In particular, it does not learn what or how much is purchased, nor when the buyer runs out of credit. While interesting, this approach is not feasible for deployment, as the customer would have to download the entire encrypted database. More recent work brings several improvements to the underlying protocols [RR01, CDN09, CDN10, HOG11], but they still only focus on *digital* goods, while our interest is in achieving customer privacy when purchasing *physical* goods.
A first step towards anonymous and pseudonymous e-commerce addresses the problem of purchasing goods with digital assets in a privacy-friendly manner [Sa14, GGM16, Go17]. Goldfeder et al. [Go17] introduce a series of escrow protocols to use when buying physical products online and paying with Bitcoin. While some of these protocols satisfy strong security properties, the buyer is still required to provide the seller with an address for delivering the goods, breaking to some extent buyer anonymity. Even though the seller does not learn the exact address of the buyer (as the address of a friend or of a post office can be provided instead), the seller learns the location where the product has to be dispatched.

3 System Overview

First, we give a brief overview of the involved parties and the relevant data.

Involved parties. The system consists of the following five parties:
- The User is a (registered) customer interested in purchasing goods online from Shop.
- Shop is the party that sells the (physical) goods through a platform accessible via Internet.
- The payment provider Pay collects the payment from User and transfers it to Shop.
- The logistics provider Shipping delivers the purchased goods from Shop to User.
- ID-Provider is a third-party responsible for managing the user's profile.

In order to prevent the Shop from collecting customers' private data and creating dossiers that reveal shopping behavior, we introduce a trusted third party in the system, ID-Provider, that increases the usability and the privacy of the architecture. It is responsible with managing the User's real and generated identities. A customer registers with ID-Provider with the real identity, and receives from ID-Provider a new generated identity, a pseudonym for logging in with Shop. Basically it acts as an authentication provider with pseudonymous identities, single sign on system for online shops and shopping process management system that connects the stakeholder for one shopping procedure. ID-Provider increases the usability for the User as well as the privacy of the overall shopping process. We require the user to provide the real identity in order to prevent system abuse. The pseudonym can be lifted in case of proved misbehavior. User can use the same pseudonym on multiple online platforms, or can create several pseudonyms, one for every platform, or even one for every purchase on the same platform.

User data. For a successful purchase, the user needs to provide the following information:

- *Product data* refers to the products selected by User for purchase.
- *Total value* refers to the purchasing price of the selected products plus additional payment and shipping charges to User.
- *Payment data* represents the data needed for a successful payment. Depending on the selected payment method this can be name, full address, bank account or credit card number, or even an anonymous payment method as sketched in Section 2. In general, banks and financial service providers require more information about a payment than just the bank account and the total value.
- *Delivery data* represents the information the delivery service Shipping needs for a successful delivery to User. In most cases, this is the name and address of User. However, other options are container freight stations and poste restante delivery, which do not necessarily require the same information.

The identifiability of the User and the linkability of purchases by Pay and Shipping depends on the chosen payment and delivery methods and applies to all architecture scenarios we will further discuss (Section 4). We assume that none of the parties collude, as collusion between Shop and any of ID-Provider, Pay or Shipping is sufficient for User profiling.

System requirements

A representation of a current e-shopping process is depicted in Figure 1. In general, Shop collects the User's data required for payment processing and package delivery. While it is possible to use a payment provider, such as Paypal [Pa17], and not provide Shop with any payment information, in most cases, the payment provider offers Shop a possibility to manage the payments and allows it to access the user's payment data.

As already discussed in Section 1, we ignore other customer tracking possibilities and focus our analysis on the data provided by User to the other parties. If a privacy-friendly online shopping platform would exist, the users could try to protect themselves via technical measures or legislation could protect the users by banning tracking without their consent.

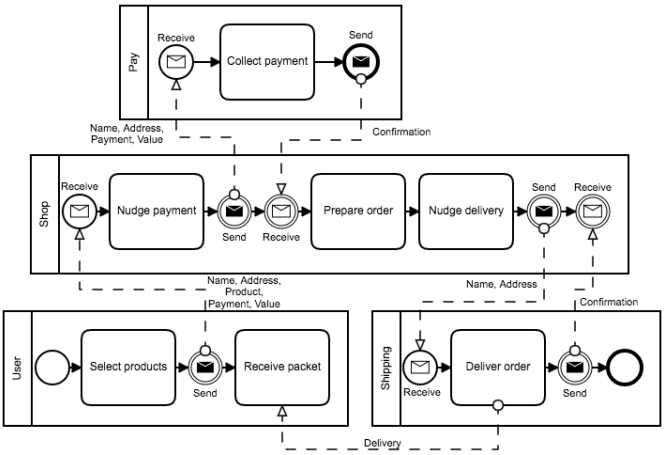


Fig. 1: Traditional Shopping Process in Business Process Modeling Notation [Ob11]

Since the login process will not differ much for the proposed architectures, the data in focus are product data, the value of the products, payment data and shipping data. When designing the pseudonymous e-commerce architecture, we aim for the *principle of minimum disclosure* under the constraints that the usability of the system should be comparable to the current systems in practice and that the process should be as transparent as possible to the user. Additionally, as discussed in Section 1, to promote a widespread use of our architecture, the shop providers business model should be respected. To chose our basic architecture, we consider the following dimensions for requirements and comparison:

Privacy. Shop should learn only the user's activity on the platform, i.e. the purchased products and their total value. The vendor does not learn payment or shipping data. Pay should learn nothing except for the amount to be payed by the user to the vendor and the payment data from the user. More specifically, the payment service does not learn the products the user purchased, but only their total value. Shipping should learn only the shipping data, but not the content (purchased products) of the package(s) to be delivered.

For the privacy analysis we apply LINDDUN, a privacy threat analysis methodology [De11, WJ15] which supports analysts in eliciting privacy requirements – similar to the security threat modeling framework STRIDE [Sh14]. Since we have a manageable number of entities in the context of our architecture scenarios, we don't run into the risk of threat explosion and can use the LINDDUN privacy analysis framework to systematically account for privacy specific threats. It is based on the graphical representation of the system's abstract representation by a data flow diagram (DFD) and the subsequent mapping of the framework's six high-level threats to each DFD element. Therefore, we model the process from checkout via payment to the delivery procedure of the products in a DFD. For each entity, we investigate the threats and map them to the elements in the DFD. In the following, we briefly discuss the privacy threats we are going to analyze.

Since the user should be able to shop pseudonymously, we consider the *identifiability* of the user (e.g. by payment or delivery data) as the main threat. In that context it is also important which parties hold which data. Depending on the party, information on *purchased products*, *the value of the purchased products*, *payment data* and *delivery data* is necessary for providing the service. As discussed, in particular the last data is suitable to revoke the User's pseudonymity. Therefore, we consider the *disclosure* of this *data* as another threat. Even if the User is not directly identifiable, *linkability* of two (or more) purchases of a user could reveal sensitive information leading to identification or at least the building of a meaningful profile. We further investigate which of the parties is able to *detect login*, *purchase*, *payment* and *delivery* events which could also be used to profile the User. Detectablility of one of the events does not mean the corresponding data is revealed, but in most cases the involved party can be identified (e.g. User did payment with Pay but neither amount nor payment data can be seen). *Unawareness* and *non-compliance* are out of the scope, as they are more related to the user interface and the entities' policies which are independent of our system architecture process. We are also not regarding *non-repudiation* for this paper since we consider it more related to contracting and legal aspects than to the architecture of our shopping platform.

Usability. Many aspects concerning the usability will not depend on the system's architecture, but on proper user interfaces allowing the user to manage his data in a easy and transparent way. However, in order to allow the user to easily use the system from different clients (e.g. computer, tablet, smart phone, ...), the user should not store information such as a cryptographic key. Additionally, the speed of the system should be comparable to existing systems, thus complex cryptographic protocols which delay the process too much can not be used. As a consequence, certain privacy enhancing technologies such as attribute based credentials [SKR12] do not come into play, because they make use of cryptographic keys, which the user would have to store on a smartcard. We compare the different architectures based on the effort the user needs to take for.

Transparency. A natural data flow which allows the user to easily understand which data is provided to whom for which purpose contributes to a transparent system. Since the user interface is out of the scope of this work, transparency of the different architectures will depend only on data flows.

Compatibility to existing business models. Analogous to attribute based credentials [Sa15], we assume that when preserving the online shop providers' business models, a broad distribution of our platform can be more easily achieved. Certainly, this does not mean that the shop providers should be allowed to collect all data they want. But allowing them to keep profiles for pseudonyms and sending e.g. newsletters (via ID-Provider) to users who gave consent would certainly be helpful for the adoption of pseudonymous e-commerce.

4 Architecture Variants

In this section we describe the three architectures, we considered for implementation. For an easier comparison, we also analyzed the current shopping process. The standard architecture allows the Shop to gather a big volume of data about its users. In order to

avoid this, we suggest three architectures, two of them make use of public-key infrastructure (see Sections 4.2 and 4.3) and a third one without encryption but self data hosting (see Section 4.4). All scenarios involve an ID-Provider for managing the user's profile.

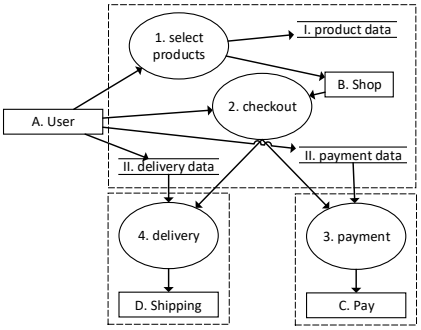
For the following analysis, we abstract from the login process and from confirmations as far as possible. Although other variants exists, we assume the User selects the products, pays and gets them delivered afterwards. Special care has to taken that Pay and Shipping providers do not pass the User's data to the Shop, e.g. by offering an administrative user interface, where payment data is listed or sending tracking information of the delivered packages. Each architecture's description follows the following template: We describe the process of every scenario and briefly discuss advantages and disadvantages. The corresponding data flow from selecting the products, checkout, payment and delivery process is depicted in Fig. 2. The analyzed privacy threats described in Section 3 are listed in Tab. 1. For each privacy threat (from Sect. 3) we denote the scenarios where it exists. For some of the analyzed threats, it depends on the users. If users don't want the shop to link their payments, they can use a new pseudonym for each purchase. For payment and shipping it depends on the kind of service the user chooses. Clearly, it makes a difference whether they are paying with anonymous electronic payment or by providing their credit card data. For shipping they could ask for home delivery or use a container freight station. We denote these threats in brackets in Tab. 1.

4.1 A: Current Shopping Process

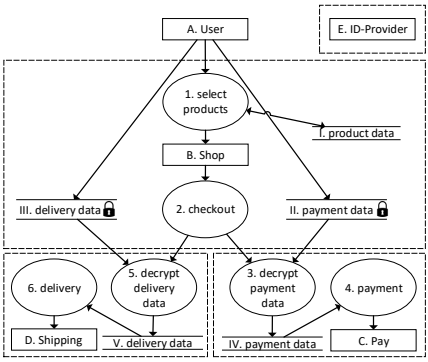
The standard shopping process is depicted in BPMN in Figure 1 and has already been described. Figure 2a shows the data flow diagram. The Shop collects all information about the user, and thus can identify the user and can link all shopping activities. The identifiability of the User and the linkability of purchases by Pay and Shipping depends on the chosen payment or delivery method. The highest privacy threat for the User is the Shop because of the possibility to disclose the User's payment and delivery data as well as profiling the User.

4.2 B: Shop Stores Encrypted Data

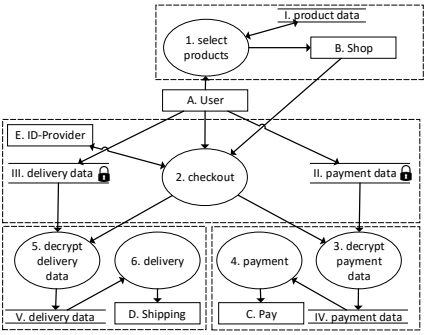
In this scenario, the user reveals only his real identity (name) to ID-Provider when registering. The ID-Provider acts as single-sign-on login service, to allow the user to log in several Shops without further registration. Additionally, ID-Provider provides public keys for payment and shipping provider. Shipping and payment data is stored encrypted on the Shop's server. The data flow of this scenario is depicted in Figure 2b.
The User initiates the process by *select products*. The Shop gets the product data and stores it. In the *checkout* process, the User decides on a payment and shipping provider. The user gets the public keys for any provider he wants to use, encrypts the payment respectively delivery data and sends it to the Shop. Subsequently, the Shop initiates the *payment* process by forwarding the encrypted banking details along with the amount to be payed to Pay. After successfully decrypting the payment data and completing the payment transaction,



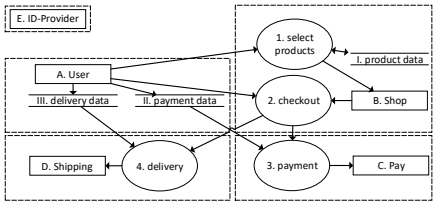
(a) Current architecture A



(b) Architecture B



(c) Architecture C



(d) Architecture D

Fig. 2: Data flow diagram

Pay sends a confirmation of payment to Shop. Upon confirmation, Shop starts the *delivery* process and sends the package labeled with the User's encrypted address to Shipping. After successfully decrypting the delivery data, Shipping proceeds with delivering the package and provides Shop with a confirmation of delivery. In this architecture Shop is not able to identify the user and can not disclose payment and delivery data. Since there is a possibility to use one pseudonym for each shopping process, the shop is also not able to link the purchases of an user unless the User allows it. The ID-Provider only knows the real identity of the user, but can not disclose payment or delivery information and also does not learn anything about the purchase process. However, the ID-Provider is able to detect the login process. The information distribution of Pay and Shipping are not affected. Therefore, the same conditions apply as for the standard architecture.

Advantages and disadvantages.

+ ID-Provider does not learn methods User uses for payment.
+ The Pay and Shipping services do not learn the virtual identity of User.
– The ID-Provider is able to detect logins in the store.
– Key management: It is difficult for the User to encrypt the payment and delivery data.

Tab. 1: Privacy Threats Mapped to Architecture Variants from Sect. 4

Threat \ Entity	Shop	Pay	Ship	Identity Provider
Identifiability	A	(ABCD)[2]	(ABCD)[3]	B C D
Disclosure shopping cart	A B C D			
Disclosure total value	A B C D	A B C D		
Disclosure payment data	A	A B C D		
Disclosure delivery data	A		A B C D	
Linkability purchase	A(BCD)[1]	(ABCD)[2]	(ABCD)[3]	C
Detectablility login	A B C D			B C D
Detectablility purchase	A B C D	A B C D	A B C D	C
Detectablility payment	A B C D	A B C D		C
Detectablility delivery	A B C D		A B C D	C

[1] Depends on the user's choice. [2] Depends on user's payment [3] Depends on user's shipping

Either this is done in the browser (e.g. with Javascript) or by an App, but the user has to trust the party providing the code.

4.3 C: ID-Provider Stores Encrypted Data

This architecture is similar to architecture B. The only change is that the (encrypted) payment and delivery data is stored at the ID-Provider. As a consequence, instead of directly delivering the data to Pay and Shipping, the Shop refers them to the ID-Provider where they need to authenticate and ask for the User's data. Therefore, the data flow itself is very similar to the one of architecture B (see 4.2) as depicted in Figure 2c. In this scenario, ID-Provider controls the shopping process. It knows the identity of the user and has information about where the user shops but does neither know the payment or delivery data since they are encrypted nor any details of the shopping content. Shop does not know the real identity of the user nor the payment or delivery details. The data distribution or possible disclosure from Pay and Shipping are unchanged.

Advantages and disadvantages.

+ ID-Provider does not learn the payment or delivery data of the User.
– If the User does not perform the encryption himself, then he has to trust ID-Provider to provide him with the correct public keys of the payment and logistics services.
– ID-Provider learns the Shop where the User makes his purchases.
– One more point of failure: ID-Provider is involved in multiple transactions.

4.4 D: User Gets Redirected to 3rd Parties

In this scenario, ID-Provider solely acts as a single-sign-on service and certification authority. All the information required for each of the steps of the pseudonymised shopping process is stored by the User. He initiates the process by *selecting the products*. The Shop gets

the product data and stores it. In the *checkout* process, the User decides on a payment and shipping provider. Subsequently, the Shop redirects the User for the *payment* process and the User delivers his payment data directly to Pay. Pay sends a confirmation of payment back to Shop. Upon confirmation, the Shop redirects the User for the *delivery* process and the User delivers his delivery data directly to Shipping. Shipping receives the package from Shop with an identifier to link it to the address and proceeds with delivering the package and provides Shop with a confirmation of delivery. Figure 2d shows the data flow. Since the User has full control about his profile data, Shop does neither know the User's identity nor the payment or delivery data. The information distribution of Pay and Shipping are not affected. Therefore, the same conditions apply as for the standard scenario.

Advantages and disadvantages.

+ The User is fully in control of his personal information.
+ Only necessary data is provided to each party.
+ Only communication needs to be encrypted.
– Additional tools have to be provided for Users to host their information.
– A lot of transactional load is put on the User. In particular, the User has to check that he is providing the information to the correct party, e.g. by checking cryptographic certificates.
– The payment process has to work instantly, otherwise additional communication is needed to synchronise payment with shipping processes.

5 Architecture comparison

In this section we compare the previously described architectures on the four dimensions described in Section 3: privacy, usability, transparency, and compatibility.

Privacy. Every involved party should learn only information about the activity belonging to its area of responsibility. In the standard scenario the Shop holds every information about the User's identity. As described in Sect. 4, all proposed architectures consider the principle of minimum disclosure. They differ only in the information provided to the ID-Provider.
In each scenario the User has the possibility to create several shopping pseudonyms. If he uses one for every shopping process, the store could not link several purchases. This applies to all architectures. The linkability of the purchase and identifiablity of the User on Pay's and Shipping's side depends on the payment and shipping methods and is not an architectural aspect. ID-Provider could link the purchases in Scenario C because ID-Provider manages the checkout process. In the other scenarios ID-Provider acts as a real identity provider and just manages the login process. Therefore, the detectability of a purchase, a payment and a delivery applies to Scenario C, but not to Scenario B and Scenario D.

Usability. Every architecture has the registration at ID-Provider and the managing of pseudonyms in common. That means compared to the standard scenario one has to maintain data not on Shop's side but on ID-Provider's side. As a compensation for managing the profiles, the User would not need to register at any Shop anymore.
Architecture B and C come with additional effort since the Users have to encrypt their data. In particular, in architecture B, Users face the problem that they might not want to

trust the Shop's App or Javascript-code making it difficult to encrypt. On the other hand in architecture C, the user has to register at the ID-Provider anyway and it seems reasonable to rely, e.g. on an App or Javascript-code on a web page. Architecture D asks the user to provide his payment and delivery data for each purchase again. This could be mitigated by making use of the The PaymentRequest API [Ba17]. However, since the recommendation is quite new, it will take some time until this has been adapted. For the authentication and single sign-on the X.509 standard could be used but needs some extensions to provide special user information. Therefore, Dash et al. [Da17] show an architecture proposal for an identity management architecture as a service. Additionally, since the Shop redirects the User to Pay and Shipping, the User has to check for each of the providers that Shop was directing her to the correct entity and not to a forged one to get the User's data.

Transparency. Despite sharing payment and delivery data directly with the Shop the standard architecture is quite transparent, because the User should be aware of sharing this data with the Shop. Although, the user might not be aware that this information might be shared with or is accessible by 3rd party service providers (e.g. webhoster, payment provider). The same holds for architecture D, where the Users need to provide their data to each entity directly. Architectures B and C, lack a bit of transparency, because it is harder for the user to assess how and from whom the encrypted data will be processed. However, it's up to the respective entity to inform the User in a supporting way.

Compatibility. The basic business processes of the involved parties are not broken by this architectures. However, by not disclosing the User's identity and therefore contact information to Shop, Shop needs to rely on ID-Provider to forward e.g. newsletters or special offers to the User. In case of misuse or disputes, ID-Provider is needed to reveal the User's identity. Pay and Shipping need to adjust their processes, in order to not reveal the User's data to Shop. However, there is no large difference here between architectures B, C, and D.

Final Architecture. Table 2 shows an overview of all attributes concerning the four analyzed architectures. While architectures B and D are favorable in respect to privacy and transparency, our focus when defining the requirements was to put emphasis on usability. Improved privacy should not complicate the shopping process for the user. The slight disadvantage in transparency from architecture C to D does not outweigh the disadvantage of architecture D that Users need to provide their data for each purchase again or alternatively have additional accounts (and logins) at payment and shipping providers. Therefore, we believe architecture C to be the most feasible option.

	Privacy	Usability	Transparency	Compatibility
A	-	o	+	++
B	++	+	o	+
C	+	++	o	+
D	++	o	+	+

Tab. 2: Comparison of the several architectures.

6 Conclusion and Future Work

In the context of pseudonymous online shopping, we presented and assessed three different architectures and compared them to the existing architecture. So far, the proof of concept shows, that a pseudonymous e-commerce process can be set up in a usable and privacy-friendly way. The User data is no longer on Shop's side but split to several parties that are involved in the shopping process.
We plan to add more processes to the shopping system such as returning goods and writing invoices. Around this, several legal and technical issues need to be resolved, e.g. how the Shop can issue an invoice to a pseudonym. Even though the PaymentRequest API only supports non-normative encryption of data fields and might also expose payments methods (cf. [Ba17, Section 19.2]), it might be helpful in storing payment and delivery data in the User's computer to avoid creating a centralized database.
Future work includes the implementation of certain restrictions for Users. For example, only Users above certain age or in certain geographical regions can access certain products. The next steps also contain the detailed description of the used protocol.

7 Acknowledgments

The SIOC project [SI] is supported by the German Federal Ministry of Education and Research's (BMBF) program "Datenschutz: selbstbestimmt in der digitalen Welt".

References

[Ac14] Acar, Gunes; Eubank, Christian; Englehardt, Steven; Juarez, Marc; Narayanan, Arvind; Diaz, Claudia: The web never forgets: Persistent tracking mechanisms in the wild. In: CCS. 2014.

[AIR01] Aiello, William; Ishai, Yuval; Reingold, Omer: Priced Oblivious Transfer: How to Sell Digital Goods. In: EUROCRYPT. 2001.

[Ba13] Bau, Jason; Mayer, Jonathan; Paskov, Hristo; Mitchell, John C: A promising direction for web tracking countermeasures. W2SP, 2013.

[Ba17] Bateman, Adrian; Koch, Zach; McElmurry, Roy; Denicola, Domenic; Cáceres, Marcos: , Payment Request API. https://www.w3.org/TR/2017/CR-payment-request-20170921/, 2017. W3C Candidate Recommendation 21 September 2017.

[CDN09] Camenisch, Jan; Dubovitskaya, Maria; Neven, Gregory: Oblivious transfer with access control. In: CCS. 2009.

[CDN10] Camenisch, Jan; Dubovitskaya, Maria; Neven, Gregory: Unlinkable priced oblivious transfer with rechargeable wallets. In: FC. 2010.

[CFN90] Chaum, David; Fiat, Amos; Naor, Moni: Untraceable electronic cash. In: CRYPTO. 1990.

[Ch83] Chaum, David: Blind signatures for untraceable payments. In: CRYPTO. 1983.

[Ch85] Chaum, David: Security without identification: Transaction systems to make big brother obsolete. CACM, 1985.

[Ch95] Chor, Benny; Goldreich, Oded; Kushilevitz, Eyal; Sudan, Madhu: Private information retrieval. In: FOCS. 1995.

[Da17] Dash, Pritam; Rabensteiner, Christof; Hörandner, Felix; Roth, Simon: Towards Privacy-Preserving and User-Centric Identity Management as a Service. In (Fritsch, Lothar;

Roßnagel, Heiko; Hühnlein, Detlef, eds): Open Identity Summit 2017. Gesellschaft für Informatik, Bonn, pp. 105–116, 2017.

[De11] Deng, Mina; Wuyts, Kim; Scandariato, Riccardo; Preneel, Bart; Joosen, Wouter: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering, 2011.

[DMS04] Dingledine, Roger; Mathewson, Nick; Syverson, Paul: Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[Ec10] Eckersley, Peter: How unique is your web browser? In: PETS. 2010.

[GGM16] Garman, Christina; Green, Matthew; Miers, Ian: Accountable privacy for decentralized anonymous payments. In: FC. 2016.

[Go17] Goldfeder, Steven; Bonneau, Joseph; Gennaro, Rosario; Narayanan, Arvind: Escrow protocols for cryptocurrencies: How to buy physical goods using Bitcoin. 2017.

[HOG11] Henry, Ryan; Olumofin, Femi; Goldberg, Ian: Practical PIR for electronic commerce. In: CCS. 2011.

[Jo16] Jourova, Vera: , How does the data protection reform strengthen citiziens' rights? http://ec.europa.eu/justice/data-protection/document/review2012/factsheets/factsheet_dp_reform_citizens_rights_2016_en.pdf, 2016.

[LRB16] Laperdrix, Pierre; Rudametkin, Walter; Baudry, Benoit: Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In: IEEE S&P. 2016.

[MM12] Mayer, Jonathan R; Mitchell, John C: Third-party web tracking: Policy and technology. In: IEEE S&P. pp. 413–427, 2012.

[Ob11] Object Management Group: , Notation (BPMN) version 2.0. OMG Specification, 2011.

[Pa17] Paypal: , Paypal Website. https://www.paypal.com, 2017.

[PCM13] Perry, Mike; Clark, Erinn; Murdoch, Steven: The design and implementation of the Tor Browser. Technical report, The Tor Project, 2013. https://www.torproject.org/projects/torbrowser/design/.

[Pr16] Preibusch, Sören; Peetz, Thomas; Acar, Gunes; Berendt, Bettina: Shopping for privacy: Purchase details leaked to PayPal. Electronic Commerce Research and Applications, 2016.

[Re16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 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) (Text with EEA relevance). Official Journal of the European Union, L 119/1, http://data.europa.eu/eli/reg/2016/679/oj, 2016.

[RR01] Ray, Indrakshi; Ray, Indrajit: An Anomymous Fair Exchange E-commerce Protocol. In: IPDPS. 2001.

[Sa14] Sasson, Eli Ben; Chiesa, Alessandro; Garman, Christina; Green, Matthew; Miers, Ian; Tromer, Eran; Virza, Madars: Zerocash: Decentralized anonymous payments from bitcoin. In: IEEE S&P. 2014.

[Sa15] Sabouri, Ahmad: Understanding the Determinants of Privacy-ABC Technologies Adoption by Service Providers. In: Open and Big Data Management and Innovation : 14th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2015. 2015.

[Sh14] Shostack, Adam: Threat modeling: Designing for security. 2014.

[SI] SIOC project website. https://sioc.eu/.

[SKR12] Sabouri, Ahmad; Krontiris, Ioannis; Rannenberg, Kai: Attribute-Based Credentials for Trust (ABC4Trust). In: TrustBus. 2012.

[St16] Statista: , E-Commerce in Deutschland. https://de.statista.com/outlook/243/137/ecommerce/deutschland/, 2016.

[WJ15] Wuyts, Kim; Joosen, Wouter: LINDDUN privacy threat modeling: a tutorial. 2015.

All websites have been last accessed on Dec. 11th, 2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 29

Anreize und Hemmnisse für die Implementierung von Privacy-Enhancing Technologies im Unternehmenskontext

Eine qualitative Analyse basierend auf Tiefeninterviews mit Privacyexperten

David Harborth[1] Maren Braun[1] Akos Grosz[1] Sebastian Pape[1] Kai Rannenberg[1]

Abstract: Wir untersuchen in diesem Artikel mögliche Anreize für Firmen Privacy-Enhancing Technologies (PETs) zu implementieren, und damit das Privatsphäre- und Datenschutzniveau von Endkonsumenten zu erhöhen. Ein Großteil aktueller Forschung zu Privatsphäre- und Datenschutz (im Weiteren *Privacy*) wird aktuell aus Nutzersicht, und nicht aus der Unternehmensperspektive geführt. Um diese bislang relativ unerforschte Lücke zu füllen, interviewten wir zehn Experten mit einem beruflichen Hintergrund zum Thema Privacy. Die Resultate unserer qualitativen Auswertung zeigen eine komplexe Anreizstruktur für Unternehmen im Umgang mit PETs. Durch das sukzessive Herausarbeiten zahlreicher Interdependenzen der gebildeten Kategorien leiten wir externe sowie unternehmens- und produktspezifische Anreize und Hemmnisse zur Implementierung von PETs in Firmen ab. Die gefundenen Ergebnisse präsentieren wir anschließend in einer Taxonomie. Unsere Ergebnisse haben relevante Implikationen für Organisationen und Gesetzgeber sowie die aktuelle Ausrichtung der Privacyforschung.

Keywords: Qualitative Tiefeninterviews; Qualitative Privacy Forschung; Privacy; Privacy-Enhancing Technologies; Firmenanreize

Also in dem Moment, wo ich sage: "Du hast hier Datenschutz und höhere Anonymität als Premium-Feature", dann hast du automatisch die Frage: „Ja, Standardkunden haben keinen Datenschutz bei euch?"

1 Einleitung

Privatsphäre- und Datenschutz (Privacy) stellen ein Grundrecht in der heutigen digitalisierten Welt dar (siehe dazu auch Datenschutz-Grundverordnung (DSGVO) der Europäischen Union [Re16]). Datenschutzfördernde Technologien (Privacy Enhancing Technologies, PETs), um diese auch umzusetzen gibt es bereits seit einigen Jahrzehnten. Allerdings werden PETs trotz technologischer Ausgereiftheit nur sehr vereinzelt verwendet [Fe01, Te17].
Dabei gibt es prinzipiell drei Akteure, die Anreize für eine entsprechende Verbreitung setzen könnten: Endverbraucher, Anbieter datenschutzbedürftiger Produkte oder Dienste

[1] Goethe University, Chair of Mobile Business & Multilateral Security, Theodor-W.-Adorno Platz 4, 60323 Frankfurt, Germany, david.harborth@m-chair.de

doi:10.18420/sicherheit2018_02

und Regulierer [Hi10, Xu12]. In bisherigen Studien wurde Privacy primär aus Perspektive der Endverbraucher untersucht [SDX11]. Ein Großteil der Endverbraucher räumt dabei anderen Faktoren als der informationellen Selbstbestimmung höhere Priorität ein. Dies zeigt sich beispielsweise an fehlender Zahlungsbereitschaft für Privacy [GA07] und daran, dass Faktoren wie Spaß die Privatsphärebedenken überlagern [DH06]. Rossnagel folgert auf Basis der Diffusionstheorie, dass Nutzer oft die Auswirkungen von PETs nicht erkennen können und deswegen für Anbieter die Vorteile der Einführung von PETs unklar sind [Ro10]. Marktwirtschaftliche Anreize, PETs einzusetzen wurden bisher für Anbieter nur in geringem Umfang untersucht. Rubinstein und der kanadische Datenschutzbeauftragte kommen dabei zum Schluss, dass aufgrund der niedrigen Nachfrage die marktwirtschaftlichen Anreize für Anbieter (oft privatwirtschaftliche Firmen) nicht groß genug sind und der Gesetzgeber Anreize schaffen sollte [Ru11, Te17]. Anreize fehlen möglicherweise auch deswegen, weil viele Geschäftsmodelle die Auswertung persönlicher Daten voraussetzen [Hu14]. Diese Strategie „verlässt“ sich zum Teil darauf, dass Anwender zu träge sind, Opt-out Optionen wahrzunehmen [Te17]. PETs, die Benutzern ein Opt-Out erleichtern würden, stehen dabei dem Geschäftsmodell entgegen.
Zusammengefasst zeigt sich, dass eine Erweiterung der Forschungsperspektive nötig ist. Die eher nutzerzentrierte Forschung muss durch Forschung aus Unternehmenssicht ergänzt werden. Es stellt sich daher die Forschungsfrage, welche Anreize und Hemmnisse Unternehmen dazu bringen bzw. davon abhalten, PETs in ihren Produkten zu etablieren.
Der Rest dieses Beitrags ist wie folgt aufgebaut: Kapitel 2 beschreibt den Forschungsstand und Kapitel 3 die verwendete Methodik. In Kapitel 4 stellen wir eine Taxonomie der Anreize und Hemmnisse für Firmen zur Einführung von PETs vor, die wir in Kapitel 5 diskutieren.

2 Aktueller Forschungsstand

Privacy-Enhancing Technologies stellt einen Sammelbegriff für verschiedene datenschutzfördernde Technologien dar. Borking und Raab definieren PETs als “coherent system of ICT measures that protects privacy [...] by eliminating or reducing personal data or by preventing unnecessary and/or undesired processing of personal data; all without losing the functionality of the data system” [BR01, S. 1]. Zusätzlich zu den PETs spielen sogenannte Transparency-Enhancing Technologies (TETs) eine wichtige Rolle dafür, dass Bürger bzw. Endverbraucher ihren Privatsphäre- und Datenschutz stärker wahrnehmen. TETs können folgendermaßen definiert werden: “[...] tools which can provide to the individual concerned clear visibility of aspects relevant to these data and the individual’s privacy” [Ha08, S. 205]. Zimmermann [Zi15] gibt einen ausführlichen Überblick am Markt existierender TETs. Die Unterschiede zwischen diesen Technologien sollen in diesem Beitrag nicht näher beschrieben werden, da es weitgehende Überlappungen zwischen ihnen gibt.
Privatsphäre- und Datenschutzthemen werden in bisherigen Studien primär aus Sicht des Individuums untersucht [SDX11]. Für unsere Forschungsfrage sind Studien interessant, die sich mit der Frage beschäftigen, inwieweit Individuen bereit sind, ihr Niveau an Privatsphäre- und Datenschutz zu erhöhen bzw. erhöhen zu lassen. Diese Fragestellung ist deshalb relevant,

da wir argumentieren, dass die Verantwortung für Privatsphäre- und Datenschutz von drei Parteien ausgehen kann, nämlich vom Individuum selbst, von Anbietern datenschutzbedürftiger Produkte oder von Regulierern. Die regulatorische Perspektive klammern wir in diesem Beitrag aus, da wir regulatorische Vorschriften mit möglichen Strafen bei Verstößen nicht als durch den Markt gegebenen Anreiz betrachten.
Forschung zu Privacy auf individueller Ebene hat gezeigt, dass Menschen angeben, sich um ihre Privatsphäre im Internet zu sorgen. Jedoch handeln sie dann entgegen ihrer vorherigen Aussagen und veröffentlichen beispielsweise zahlreiche persönliche Informationen in sozialen Netzwerken. Aktuelle Forschung erklärt dieses Verhalten einerseits mit einer Art kognitiver Dissonanz, die beim Thema Privacy hervortritt (vgl. Privacy Paradox [NHH07, SGB01]). Dieser Erklärung entgegenstehend sehen zahlreiche andere Forscher einen bewussten Trade-Off im Sinne eines Austausches eines speziellen Nutzens (kostenfreie Dienstleistung, Anerkennung, etc.) gegen Daten, auf den Nutzer sich einlassen (vgl. Privacy Calculus [DH06, DT15, DM16]). Weitere Forschung zeigt, dass Individuen neben dieser Divergenz von geäußerter Einstellung und beobachtbarer Handlung nur wenig Kosten (zeitlich und monetär) für ihre Privatsphäre tragen möchten [GA07].
Insbesondere der letzte Punkt wirft die Frage auf, inwieweit es unter diesen Voraussetzungen möglich ist, PETs profitabel am Markt zu etablieren. Daher ist es relevant, die Unternehmensperspektive auf Anreize für Unternehmen zu beleuchten. Die bisherige Forschung in diesem Gebiet ist nicht so reif wie im Gebiet der Forschung zu Privacy und Individuen [SDX11]. Einige der Artikel beschäftigen sich mit den Konsequenzen von Privatsphäre- und Datenschutzverletzungen in Firmen [AFT06] und wie Firmen mit diesen Verletzungen umgehen können [CKJ16]. Relativ viele Beiträge untersuchen, inwieweit Privacy ein kompetitiver Vorteil ist und sich in Geschäftsmodelle integrieren lässt [Ho14, CMHD15, Li11].

3 Methodik

In diesem Kapitel besprechen wir die verwendete qualitative Forschungsmethodik. Wir folgen diesem explorativen Ansatz, da bisherige Forschung unsere Forschungsfrage unzureichend adressiert hat. Im ersten Schritt haben wir einen semi-strukturierten Leitfadenfragebogen entworfen. Basierend auf dem semi-strukturierten Fragebogen werden die Teilnehmer durch das Interview geführt. Semi-strukturiert bedeutet in diesem Zusammenhang, dass das Interview maßgeblich durch die Interaktion und die Antworten des Befragten beeinflusst wird. Der Fragebogen hält nur besonders relevante Fragen fest, die auf jeden Fall angesprochen werden wollen. Dies hat den Vorteil, möglichst tiefe Einblicke und ausführliche Antworten vom Teilnehmer erhalten zu können. Der Fragebogen kann in drei inhaltliche Oberthemen aufgeteilt werden. Zuerst werden allgemeine Fragen zur Person und zum Unternehmen gestellt. Darauf folgen Fragen zu Privacy und PETs. Der zweite Teil deckt technische Fragen zum Status Quo und zu eventuellen zukünftigen Entwicklungen ab. Der dritte Teil behandelt ökonomische und gesellschaftliche Fragestellungen.
Für die Beantwortung unserer Forschungsfrage haben wir Experten und Professionals befragt, die mit Privacy-Enhancing Technologies (PETs) in ihren Unternehmen zu tun haben,

oder bei deren Produkten oder Dienstleistungen Privacy eine besondere Rolle spielt. Die Experten stammen von Firmen, die direkt PETs anbieten, oder in denen Privacy eine wichtige Rolle im Nutzenversprechen spielt. Als Beispiele hierfür sind der Telekommunikationssektor, Paymentprovider oder eCommerce Solutions Provider zu nennen. Wir haben zehn Interviews geführt und analysiert, wobei die Dauer zwischen 44 Minuten und 180 Minuten variiert. Die demografischen Informationen finden sich in Tabelle 2.
Die Interviews wurden alle aufgezeichnet und anschließend Wort für Wort transkribiert. Die Transkriptionen wurden daraufhin mit dem sog. offenen Kodieren und selektiven Kodieren analysiert [GS67, Ch14, St13]. Das offene Kodieren ist der erste Schritt der Datenauswertung und orientiert sich nah an den Daten (den Transkripten). Im nächsten Schritt werden Kodes zusammengefasst und abstrahiert (selektives Kodieren). Diese Schritte werden für jedes Interview einzeln durchgeführt und anschließend zwischen den Interviews. Diese sogenannte komparative Methode [GS67, Ch14, St13] ist ein elementarer Bestandteil der qualitativen Forschungsmethodik. Durch ständiges Vergleichen zwischen den Interviews, leiten wir abstrakte Kategorien aus den Daten ab, die ein vielfältiges Bild der Anreize und Hemmnisse liefert. Diese Kodierungsschritte wurden von zwei Autoren durchgeführt, um eventuelle Diskrepanzen in der Analyse der Daten festzustellen und zu lösen.

4 Resultate

Wir stellen in diesem Abschnitt das Kategoriensystem vor, welches elementar wichtig für eine logische und aufeinander aufbauende Strukturierung der Ergebnisse ist. Unsere übergeordnete Zielsetzung lag bei diesem Prozess darin, die Unternehmensperspektive in Bezug auf PETs nachzuvollziehen und zu verstehen. Da die Interviewteilnehmer sehr vielschichtige, sowohl vergleichbare und aufeinander aufbauende, als auch gegensätzliche, Stellungen bezogen, leitete sich hieraus eine argumentative Gliederung der relevanten Themenkomplexe ab. Diverse Querbezüge und Wechselwirkungen markieren damit ein interdependentes Gefüge, in welchem Unternehmen Anreize und Hemmnisse für die Implementierung von PETs betrachten. Ein Interviewteilnehmer fasst diese komplexen unternehmenszentrierten Abwägungsentscheidungen im Rahmen des Einsatzes von Privacy-Enhancing Technologies in den folgenden Worten zusammen: "*Ja, es [meint: PETs] soll funktionieren. Ja, und die, die es betreiben, sollen davon leben können. Ja, das soll so sein, aber es soll so sein, dass eben die Kontrolle, Transparenz und Nutzbarkeit breit akzeptierbar ist*" (D).

4.1 Technische Optimierung

Der Großteil der Interviewpartner gab an, dass PETs dienlich sind, um allgemein Unternehmensprozesse auf technischer und organisatorischer Ebene zu optimieren, "*dass man eben vor allem einen technologischen Vorsprung hat*" (B). Die spezifische Modellierung und Funktionalität der Technologie fördert dabei, dass Abläufe im Unternehmen unterstützt, vereinfacht und auch bedarfsgerecht angepasst werden können, was einen technologischen Ansatz zur Realisierung von Privacy-Maßnahmen im Unternehmen darstellt.

Tab. 1: Taxonomie

1. Technische Optimierung	1. Integration in den Geschäftsprozess	
	2. Datenmanagement und -vermeidung	
2. Geschäftsmodell	1. Weiterentwicklung Services	1. Kundenanforderung
		2. Vereinfachung / Convenience
		3. Awareness / Visualisierung
	2. Erweiterung Kundenkreis	1. Kerngruppe mit Privacybedürfnis
		2. Massenmarkt und Segmentierung
	3. Entwicklung neuer Geschäftsmodelle	1. Premiumservice
		2. Wirtschaftlichkeitsabwägung
	4. Positionierung für die Zukunft	
3. Unternehmens-wahrnehmung	1. (Technische) Sicherheit	
	2. Profilierung durch PETs	
	3. Geschäftsethik	

Integration in den Geschäftsprozess. Notwendige Bedingung für die Erwägung einer PET-Implementierung in den Geschäftsprozess ist, inwiefern Tools auf technischer Ebene auf relevante Prozesse abbildbar sind bzw. ob der Kostenaufwand in einem für angemessen erachteten Rahmen liegt: "*Gibt es so etwas? Wieso brauche ich so etwas? Wie kriege ich so etwas? Wie installiere ich so etwas? Und dann, wie setze ich es richtig für meinen Gebrauch ein?*" (H). Das Fehlen einer gesicherten Informationsgrundlage diesbezüglich wurde allerdings bemängelt: "*Man kann sich Vieles vorstellen, ob es dann in der Realität umsetzbar ist, ist dann die Frage*" (A).

Datenmanagement und -vermeidung. PETs können ein vereinfachtes und adäquates Datenmanagement gewährleisten, um darüber hinaus auch für den Geschäftsprozess nicht notwendige Daten vermeiden zu können. Dies kann letztlich auch mit dem gänzlichen Verzicht personenbezogener Daten einhergehen. Ein zentraler Anreiz für die Implementierung von PETs ist daher, dass Unternehmen die unmittelbare Entscheidungshoheit über Erhebung und Aggregation von Daten erlangen. Ein Interviewpartner erörtert, dass "*man immer von irgendwelchen Daten irgendwelche Rückschlüsse ziehen kann*" (A), weshalb Daten außerhalb ihrer jeweiligen Nutzung und Notwendigkeit als ein zusätzliches Geschäftsrisiko bewertbar sind. Ein weiterer Interviewpartner leitet aus der Vermeidung von Daten einen positiven Nutzen für Unternehmen ab: "*Wenn die Daten zum Beispiel nur dort sind, wo sie überhaupt gebraucht werden, dann brauche ich da nicht irgendwie auf den anderen Systemen, wo sie nicht gebraucht werden, erstmal Verschlüsselungen und Maßnahmen [mit]ergreifen*" (F). Die Möglichkeit auf schlankere und einfachere Unternehmensabläufe wurde ferner ebenfalls herausgearbeitet. Andererseits bieten unverschlüsselte personenbezogene Daten den Vorteil, eindeutig einem angelegten Profil und jeweiligen geschäftlichen Aktivitäten zugeordnet werden zu können: "*[Ein Klarname] ist natürlich einfacher, um Transaktionen zuzuordnen, um Versand beispielsweise einem gewissen Kunden zuzuordnen. Aber grundsätzlich wäre das auch über ein Pseudonym schon machbar*" (B).

4.2 Geschäftsmodell

Die Kategorie Geschäftsmodelle stellte sich im Rahmen unserer Auswertung als umfangreichste Kategorie dar (Schlüsselkategorie). Auf dieser Ebene wurden sowohl die stärksten Anreize als auch Hindernisse an die Forschenden herangetragen. Die zahlreichen Freiheitsgrade und Gestaltungsoptionen in Zusammenhang mit PETs wurden als primär ausschlaggebend für die hohe Schwingungsbreite des erwarteten Geschäftserfolgs bewertet. Der vorliegende Status quo wurde von einem Interviewpartner durchaus auch optimistisch aufgefasst: "*Wir können mit [PETs] glaube ich völlig neue Geschäftsmodelle aufbauen, die der Markt bisher überhaupt noch nicht kennt. [...] Wir wollen tatsächlich ein paar Schuhe von A nach B bringen anonym. Das kann aber auch was völlig anderes sein*" (E).

Weiterentwicklung Services. Unsere Ergebnisse zeigen, dass die Implementierung von PETs auch dazu beitragen kann, dass Unternehmen bestehende Services weiter in Richtung Datenschutz entwickeln können und damit in spezifischen Marktstrukturen einen Wettbewerbsvorteil generieren können.

KUNDENANFORDERUNG. Wie stark Kundenanforderungen hinsichtlich des Privatsphäre- und Datenschutzniveaus ausgeprägt sind, steht in Verbindung damit, welche Kundenstruktur gegeben ist und auf welche Segmente künftig spekuliert wird. Aus den Interviews ging sowohl hervor, dass es Kunden gibt, die ein großes Interesse hegen, sich zu schützen und sich diesbezüglich mit Nachfragen wie Ansprachen an Unternehmen wenden, als auch die Auffassung der Interviewpartner, dass Privacy keine bzw. eine eher untergeordnete Rolle beim Gros der (potenziellen) Kunden spielt. Eine etwas andere Konnotation sehen wir indes darin, dass Kunden einen ausreichenden Privatsphäre- und Datenschutz selbstredend erwarten, dies allerdings nicht zwingend explizit an Unternehmen herangetragen: "*Da erwartet der Kunde auch, dass da gewisse Schutzmechanismen passieren. [...] Und das passiert auch. Und das bezahlen sie auch implizit*" (D).

VEREINFACHUNG UND CONVENIENCE. "*Jetzt speziell auf das Internet gesehen, [. . .] jeder gibt irgendwie Daten dort preis. [...] Ist halt oft schwer, nur das preiszugeben, was man möchte, weil man eben doch oft Dinge preisgibt, von denen man nicht weiß oder in dem Moment, wo man sie preisgibt, eben nicht weiß, was damit geschieht letztendlich*" (B). Im Anschluss an diesen Problemaufriss fassen wir unter dieser Kategorie, dass (mögliche) Kunden unbefangen eine Geschäftsbeziehung mit einem Anbieter eingehen und aufrechterhalten können, da entsprechende Services hinsichtlich ihrer Privacy-Einstellungen verbessert wurden. Den Konsumenten wird dadurch kommuniziert, dass sensible Daten nicht erhoben bzw. diese ausreichend durch jeweilige Mechanismen geschützt werden: "*[Die Kundenansprache] könnte man jetzt so machen: '[...] Wir haben jetzt eine neue Technologie [...] installiert [...] und die Möglichkeit schützt deine privaten Daten. Sonst bleibt für dich alles gleich.' [...] Es ist leicht verständlich. Er muss die Technologie auch nicht verstehen*" (A). Andererseits betrachtete ein Teil der Interviewpartner Einfachheit und Bequemlichkeit unter dem Gesichtspunkt, dass datenschutzfördernde Tools aus ihrer Sicht eher einen Mehraufwand für Kunden darstellen: "*Letztendlich, warum wir die Daten speichern möchten oder teilweise speichern wollen, ist eben, um dem Kunden zu vereinfachen, dass er beim nächsten Mal zum Beispiel dann nichts mehr eingeben muss. [...] Also letztendlich ist*

das immer so eine Abwägung zwischen Privatsphäre und zu viele Daten sammeln oder Einfachheit, also im Grunde eben ein möglichst einfaches Interface zu bieten" (B).

AWARENESS UND VISUALISIERUNG. Unternehmen wurde eine wichtige Rolle dabei zugesprochen, auf die Privacy-Thematik aufmerksam zu machen und (potenzielle) Kunden hierfür zu sensibilisieren. Die Interviewteilnehmer erachteten dies als angemessene Maßnahme, um Nachfrage auf diesem Gebiet zu generieren. Gleichzeitig sah ein Befragter Firmen in dieser Hinsicht nicht in der Verantwortung: "*Wir brauchen Awareness von den verschiedenen Segmenten, die es brauchen*" (H). Weiterhin wurde eine geeignete Form der Visualisierung als substanziell eingestuft, um Nutzern die Vorteilhaftigkeit von PETs vor Augen zu führen: "*Irgendwo hätte ich schon gerne als Endteilnehmer, wenn ich schon bezahle, ja, warum bezahle ich eigentlich? Also da muss irgendwie so eine Beweisnotwendigkeit sein*" (D). Inwiefern ein Premium- oder Upselling-Preismodell als sinnvoll zu erachten ist und ob ein Zusatznutzen wie die genannte Visualisierung einen attraktiven Trade-off für Kunden darstellt, werden wir im Abschnitt "Premiumservice" näher diskutieren.

Erweiterung Kundenkreis. Eine Privacy-freundliche Ausrichtung von Unternehmen kann neue Kundenmärkte öffnen und ein Alleinstellungsmerkmal darstellen.

KERNGRUPPE MIT PRIVACY-BEDÜRFNIS. Die Erweiterung des Kundenkreises ist ein häufig von den Befragten formulierter Anreiz für Unternehmen, PETs zu implementieren. Im Kern fielen darunter Personen, bei denen ein Privacy-Bedürfnis bereits überdurchschnittlich stark ausgeprägt ist: "*Ich adressiere genau diese Lücke. Ich adressiere die Freaks, ich adressiere die Nerds, ich adressiere diejenigen, die mehr Privacy Awareness haben, als die anderen*" (G). Nicht nur technikaffine und -interessierte Privatpersonen sind für die Befragten Teil dieser Kategorie, sondern auch Forschungs- und Entwicklungszentren sowie bestimmte Unternehmen. Neben intrinsischen Motiven PETs nachzufragen, spielen für Geschäftskunden häufig auch rechtliche Vorgaben zum Datenschutz eine zentrale Rolle.

MASSENMARKT UND SEGMENTIERUNG. Interviewteilnehmer lieferten sehr nuancierte Aussagen, bezüglich der möglichen Eignung von PETs im Massenmarkt (Primär- und Sekundärnutzen betrachtend). Wir haben diese vielfältigen Blickwinkel aufgegriffen, da sie im Rahmen ihrer jeweiligen Logik nicht zwingend als Widerspruch zu betrachten sind. Ein Potenzial zum Massenmarkt wurde unter anderem beschrieben, um ein besonders hohes Privatsphäre- und Datenschutzniveau als wünschenswerten Idealzustand in den Vordergrund zu stellen. Ein Befragter gab in diesem Zusammenhang an, dass "*jeder, der eine Kundenbeziehung hat*" (D), PETs im Sinne seiner Kunden implementieren sollte. Es ließ sich zudem der Konsens herauslesen, dass dies vom Großteil der Nutzer nicht explizit gefordert und nachgefragt wird, sondern eher akzeptiert, dann aber auch als positiver Nutzen empfunden wird: "*'Kann ich da Datenschutz einschalten? Ja oder nein?' Und dann glaube ich schon, dass viele Leute sagen: 'Joa, einschalten. Datenschutz ist immer gut'*" (G). Des Weiteren wurde Massentauglichkeit darin gesehen, dass PETs in bereits bestehende Produkte als Sekundärnutzen implementiert werden: "*Welche PETs setzen sich bisher im Massenmarkt durch? Nur eigentlich in Begleitung mit anderen Services eben*" (C). Als integraler Baustein etablierter Leistungen können PETs dadurch gar ohne aktives Nutzereinverständnis und ohne konkrete Nachfrage auf dem Massenmarkt etabliert werden. Gleichzeitig wurde die Notwendigkeit angeführt, in Marktsegmente zu unterscheiden: "*Seit Jahren habe ich*

gesagt, argumentiert, dass Mass Marketing ein Fehler ist. [...] Die Unterschiede zwischen den Anforderungen von den verschiedenen Segmenten [...] sind so groß. Verschiedene Leute brauchen unterschiedliche Unterstützung" (H). Diese Aussage steht in Verbindung damit, dass im Rahmen der geführten Interviews zahlreiche potenzielle Nutzergruppen genannt wurden: Unternehmen verschiedener Größe, Forschungsinstitutionen, öffentliche Einrichtungen, Privatpersonen, für die der Schutz der Privats- und Intimsphäre von hoher Bedeutung ist, beispielsweise, wenn sie "*in einem speziellen Segment sensible Produkte*" (A) erwerben möchten. Eine dritte Gruppe von Befragten bewertete einen größeren Kundenkreis hingegen als unrealistisch: "*Ich denke, dass es [meint: PETs] in gewisser Weise schon auch ein Nischenmarkt ist, also dass es nicht unbedingt massenmarkttauglich ist, weil einfach zu vielen Menschen die Privatsphäre da zu unwichtig ist [. . .] beziehungsweise unwichtig genug, um keine extra Mühen auf sich zu nehmen*" (B).

Entwicklung neuer Geschäftsmodelle. Neben der Erschließung neuer Kundenmärkte, kann eine datenschutzfreundlichere Ausrichtung neue Geschäftsmodelle ermöglichen.

PREMIUMSERVICE. Die Interviewten hatten keine eindeutige Meinung, inwiefern sich durch die Implementierung einer PET ein Premium- oder Upselling-Service geschäftlich sinnvoll ist. Ein Befürworter dieses Preismodells erklärte: "*Ja, es [meint: PETs] kostet was. Das ist wieder der berühmte Punkt: Es gibt etwas kostenlos, dann ist es aber eine mildtätige Spende, wo jemand sagt: 'Jawohl, ich spende das dafür, dass es auch wirklich kostenlos ist, so.' Alle anderen Sachen haben irgendwo ihren Trade-off. [. . .] Wir können genau sagen: 'Das kostet es, das bringen wir. Macht mit oder lasst es bleiben'*" (D). Allerdings können bestehende Leistungen des Unternehmens, die eventuell um keine datenschutzfördernde Technologie erweitert wurden, degradiert werden: "*Du versuchst ein Premium-Feature zu positionieren, aber gleichzeitig qualifizierst du alle anderen [angebotenen Services] ab und bringst die in eine Situation, dass du dich für die dann rechtfertigen musst: 'Warum kriegen das nicht alle?' Und die zweite Frage ist: Wer ist bereit für ein solches Premium-Feature zu bezahlen? Es ist dann irgendwas Exklusives*" (G). Daran anknüpfend bezieht ein Interviewteilnehmer folgende Position: "*Die monetären Kosten muss [das Unternehmen] kalkulieren*" (F).

WIRTSCHAFTLICHKEITSABWÄGUNG. Ohne Premium-Services müssen Unternehmen die Kostendeckung einer PET-Implementierung anderweitig garantieren, zum Beispiel durch Absatzsteigerung: "*Wir würden nur über Mengensteigerungen verdienen, weil wir dieses System anbieten*" (A). Alternativ ist es auch möglich, die Konversionsrate durch PETs zu steigern: "*Dem Hersteller nutzt es dann, wenn die Kunden einen Nutzen dahinter sehen und wenn es vielleicht dieser winzige Ausschlag ist, der eine Kaufentscheidung beeinflusst*" (G). Neben diesen quantifizierbaren Aspekten spielen weitere Faktoren, z.B. eine heuristisch orientierte Kosten-Nutzen-Analyse, eine zentrale Rolle in den jeweiligen Abwägungsentscheidungen. Diese gehen mit einer negativen Konnotation von Datenerhebungsvermeidung einher, bspw. durch Betrugsfälle: "*Ich denke [Unternehmen] werden auf jeden Fall erstmal Vorbehalte gegen so etwas [meint: PETs] haben, eben dadurch, dass sie befürchten, für irgendwelche Betrugsfälle oder so keinen greifbaren Kontakt irgendwie zu haben*" (B). Zum anderen wurde mehrfach folgende Befürchtung akzentuiert: "*Die [Unternehmen] haben natürlich kein Interesse an einem Pseudonym, denn die wollen ja Daten, Profile, Bewegungsprofile erstellen, weil das bares Geld ist*" (I).

Positionierung für die Zukunft. Zum einen betonten Befragte die Möglichkeit des Alleinstellungsmerkmales von PETs: "*Das wäre das Alleinstellungsmerkmal irgendwie für uns auch, [ein Produkt] eben anzubieten, [das] die Identität des Kunden schützt, was es eben zurzeit noch nicht so gibt, ja, also vor allem eben auch irgendwie dadurch einen Wettbewerbsvorteil zu gewinnen*" (B). Allerdings schwindet dieser Vorteil eventuell, wenn eine kritische Masse an Wettbewerbern ebenfalls vermehrt PETs implementieren oder Wettbewerber mit bedeutenderer Marktmacht bestimmte Schutztechnologien als neuen „Standard" etablieren: "*Die [meint: https-Verschlüsselung] setzt sich durch, langsam, weil tatsächlich große Konzerne auch dahinter stehen und das jetzt auch forcieren*" (C). Zum anderen wurde PETs eine präventive Wirkung beigemessen, um „Datenschutzskandale" zu vermeiden: "*Man [hat] das schon noch natürlich immer im Hinterkopf, weil man irgendwie auch ganz sicher nicht das Unternehmen sein möchte, was irgendwie in den Schlagzeilen ist, jetzt irgendwie auffällt dadurch, das die Privatsphäre nicht schützt.*" (B).

4.3 Unternehmenswahrnehmung

Privatsphäre- und Datenschutztechnologien verfügen über das Potenzial, sowohl die externe als auch die interne Wahrnehmung des Unternehmens zu beeinflussen.

(Technische) Sicherheit, Vertrauen und Qualität. Für das Vertrauensverhältnis zwischen Geschäftspartnern spielt das Verständnis der jeweiligen Technologie nur eine sekundäre Rolle. Die positive Wahrnehmung entstammt vorranging der impliziten Gefühlsebene: "*Heute verkaufen sich Sachen gut [...] indem gesagt wird: 'Wir machen das nach deutschen Datenschutzrechten [...].' Das verstehen die Leute. Die kennen überhaupt null Details dazu, aber die sagen sich: 'Okay. Wenn das nach deutschem Datenschutzding ist, dann passt das'*" (E). Die durch PETs gewährleistete Vertrauensfestigung kann sich dabei positiv auf den Ruf des Unternehmens niederschlagen: "*Ich sehe es als Qualitätsmerkmal*" (E).

Profilierung durch PETs. Die Kopplung von PETs an das bekannte Dienstangebot des Unternehmens stellt zudem ein kommunizierbares Alleinstellungsmerkmal dar, wie einer der Befragten erläuterte: "*Man kann es als Werbezweck verwenden. [...] Ich unterscheide mich damit von anderen. Das muss jetzt nicht sein, dass das so einen wahnsinnigen Zusatznutzen hat, es ist einfach ein Marketing-Effekt, den ich damit verbinden kann*" (C). Dieser allgemeine Werbeeffekt fördert dann nicht nur das reguläre Angebot, sondern auch die Reputation des Unternehmens: "*Ich glaube du kannst es [meint: mit PET-Implementierung auch Profitabilität sichern] nur machen, wenn du das als Add-on zu deinem Produkt [anbietest]. [...] Dann sagst du: 'Okay, ich investiere jetzt halt mal, weil das bringt mir vielleicht etwas in meinem Ansehen, in meinem Ruf, in meiner Zahl der [Kunden]'*" (I).

Geschäftsethik. Drei ethische Momente des unternehmerischen Handelns heben sich im Hinblick auf Privatsphäre- und Datenschutztechnologien aus den Interviews hervor. Erstens wird die These angeführt, dass Technologien und ihre Nutzung unabhängig von moralischen Wertepositionen gegeben sind: "*Es existiert, es ist keine Frage, keine moralische Frage. Es gibt Anonymität, das ist ein Konzept und es kann für verschiedene Zwecke benutzt werden*" (H). Diese an sich neutrale Auffassung von PETs kann polarisierenden Darstellungen gegenübergestellt werden. So können sich daraus moralisch vertretbare

Schritte zur informativen Sensibilisierung ergeben, jedoch auch verwerfliche, wie zum Beispiel eine einseitige, überspitzte Beängstigungskampagne. Diese können sich gar als geschäftsschädlich herausstellen: "*Natürlich, ich nutze es, ich habe Angst, aber ich weiß auch, dass ich das Produkt nur nutze, weil ich Angst habe. Es macht es jetzt auch nicht unbedingt so wahnsinnig sympathisch*" (C). Letztlich stehen moralische Aspekte der ökonomischen "Rationalität" von Firmen entgegen: "*Ich investiere jetzt in etwas und [...] ich mache das erst einmal, weil ich der Meinung bin: 'Das ist richtig und es hilft und es ist das Richtige zu tun und langfristig profitiere ich vielleicht auch davon, vielleicht nicht finanziell.' Das macht kein Unternehmen*" (I).

5 Diskussion und Schluss

Basierend auf der qualitativen Auswertung von zehn Tiefeninterviews mit Privacyexperten haben wir eine Taxonomie der Anreize und Hemmnisse für die Implementierung von PETs im Unternehmenskontext entwickelt.
Gemäß der Taxonomie spielen die mit Geschäftsmodellen verbundenen Anreize eine wichtige Rolle. Wie bestehende Literatur kommen wir allerdings zum Schluss, dass es Bedarf für weitere Forschung in dem Bereich zu Privatsphäre- und Datenschutz speziell im Unternehmenskontext gibt. Beispielsweise argumentiert Rubinstein [Ru11], dass die marktwirtschaftlichen Anreize für Firmen nicht gross genug sind und eine flächendeckende Verbreitung von PETs nur aufgrund von Initiativen des Gesetzgebers stattfinden wird. Ein weiteres vielversprechendes Thema für zukünftige Forschung besteht in dem Vergleich von Evaluierungen und Meinungen verschiedener Privacyexperten. Unsere Ergebnisse zeigen in einigen Bereichen kein klares Bild, da die Aussagen teilweise weit auseinander gehen. Befragte haben einerseits sehr unterschiedliche berufliche (Unterschiede in Firmen bezüglich Marktumfeld und Marktgröße) und private Hintergründe und andererseits sind ihre Positionen entweder ethisch oder praxisorientiert.
Wir tragen zur aktuellen Privacyforschung auf drei Wegen bei. Erstens haben wir Privacy im Unternehmenskontext, und nicht auf individueller Ebene, untersucht [SDX11]. Zweitens haben wir eine empirische, nicht normative, Studie durchgeführt, die auf einem Sample mit deutschen Interviewteilnehmern basiert. Zum Großteil ist Privacyforschung normativ und basiert auf Stichproben mit US-amerikanischen Teilnehmern [BC11]. Drittens haben wir mit einer qualitativen Methodik ein unterrepräsentiertes Thema explorativ von verschiedenen Dimensionen erforscht. Zusammenfassend zeigen unsere Ergebnisse, dass es durchaus Anreize für Unternehmen (abgesehen von Regulierung) geben kann, datenschützfördernde Technologien und Strukturen in ihren Geschäftspraktiken zu implementieren und damit dem Datenschutz zukünftig mehr Relevanz zu geben.

6 Acknowledgments

Diese Forschung wurde vom Bundesministerium für Bildung und Forschung (BMBF) unterstützt (Zuwendungsnummern 16KIS0371 and 16KIS0515).

Literaturverzeichnis

[AFT06] Acquisti, Alessandro; Friedman, Allan; Telang, Rahul: Is There a Cost to Privacy Breaches? An Event Study. In: International Conference on Information Systems (ICIS). 2006.

[BC11] Bélanger, France; Crossler, Robert E.: Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems. MIS Quarterly, 35(4):1017–1041, 2011.

[BR01] Borking, John J.; Raab, Charles: Laws, PETs and Other Technologies for Privacy Protection. Journal of Information, Law and Technology, 1:1–14, 2001.

[Ch14] Charmaz, Kathy: Constructing Grounded Theory. Sage Publications, London, 2nd editio. Auflage, 2014.

[CKJ16] Choi, Ben C.F.; Kim, Sung S.; Jiang, Zhenhui (Jack): Influence of Firm's Recovery Endeavors upon Privacy Breach on Online Customer Behavior. Journal of Management Information Systems, 33(3):904–933, 2016.

[CMHD15] Casadesus-Masanell, Ramon; Hervas-Drane, Andres: Competing with Privacy. Management Science, 61(1):229–246, 2015.

[DH06] Dinev, Tamara; Hart, Paul: An extended privacy calculus model for e-commerce transactions. Information Systems Research, 17(1):61–80, 2006.

[DM16] Dienlin, Tobias; Metzger, Miriam J.: An Extended Privacy Calculus Model for SNSs: Analyzing Self-Disclosure and Self-Withdrawal in a Representative U.S. Sample. Journal of Computer-Mediated Communication, 21(5):368–383, 2016.

[DT15] Dienlin, Tobias; Trepte, Sabine: Is the privacy paradox a relic of the past? An in-depth analysis of privacy attitudes and privacy behaviors. European Journal of Social Psychology, 45(3):285–297, 2015.

[Fe01] Feigenbaum, Joan; Freedman, Michael J; Sander, Tomas; Shostack, Adam: Privacy engineering for digital rights management systems. In: Digital Rights Management Workshop. Jgg. 2320. Springer, S. 76–105, 2001.

[GA07] Grossklags, Jens; Acquisti, Alessandro: When 25 Cents is Too Much: An Experiment on Willingness-To-Sell and Willingness-To-Protect Personal Information. In: WEIS. 2007.

[GS67] Glaser, Barney G.; Strauss, Anselm L.: The Discovery of Grounded Theory. Aldine Pub., Chicago, 1967.

[Ha08] Hansen, Marit: Marrying Transparency Tools with User-Controlled Identity Management. In (Fischer-Hübner, S.; Duquenoy, P.; Zuccato, A.; Martucci, L., Hrsg.): The Future of Identity in the Information Society. IFIP — The International Federation for Information Processing, S. 199–220. Springer, Boston, MA, 2008.

[Hi10] Hirsch, Dennis D: The law and policy of online privacy: Regulation, self-regulation, or co-regulation. Seattle UL Rev., 34:439, 2010.

[Ho14] Hoffman, David: Privacy Is a Business Opportunity. Harvard Business Review, S. 2–5, 2014.

[Hu14] Hustinx, Peter: Preliminary Opinion of the European Data Protection Supervisor "Privacy and competitiveness in the age of big data: The interplay between data protection, competition law and consumer protection in the Digital Economy". Bericht, European Data Protection Supervisor, March 2014. available via https://edps.europa.eu/sites/edp/files/publication/14-03-26_competitition_law_big_data_en.pdf.

[Li11] Liu, Zhan; Bonazzi, Riccardo; Fritscher, Boris; Pigneur, Yves: Privacy-friendly business models for location-based mobile services. Journal of Theoretical and Applied Electronic Commerce Research, 6(2):90–107, 2011.

[NHH07] Norberg, Patricia A.; Horne, Daniel R.; Horne, David A.: The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors. Journal of Consumer Affairs, 41(1):100–126, jun 2007.

[Re16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 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) (Text with EEA relevance). Official Journal of the European Union, L 119/1, http://data.europa.eu/eli/reg/2016/679/oj, April 2016.

[Ro10] Rossnagel, Heiko: The Market Failure of Anonymity Services. In (Samarati, Pierangela; Tunstall, Michael; Posegga, Joachim; Markantonakis, Konstantinos; Sauveron, Damien, Hrsg.): Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices: 4th IFIP WG 11.2 International Workshop, WISTP 2010, Passau, Germany, April 12-14, 2010, Proceedings. Springer, 2010.

[Ru11] Rubinstein, Ira S: Regulating privacy by design. Berkeley Technology Law Journal, 26(3):1409–1456, 2011.

[SDX11] Smith, H. Jeff; Dinev, Tamara; Xu, Heng: Theory and Review Information Privacy Research: An Interdisciplinary Review. MIS Quarterly, 35(4):989–1015, 2011.

[SGB01] Spiekermann, Sarah; Grossklags, Jens; Berendt, Bettina: E-privacy in 2nd generation E-commerce. In: Proceedings of the 3rd ACM conference on Electronic Commerce - EC '01. ACM Press, New York, New York, USA, S. 38–47, oct 2001.

[St13] Strübing, Jörg: 1978. 2013.

[Te17] Technology Analysis Division of the Office of the Privacy Commissioner of Canada: Privacy Enhancing Technologies - A Review of Tools and Techniques. Bericht, Office of the Privacy Commissioner of Canada, November 2017. available via https://www.priv.gc.ca/en/opc-actions-and-decisions/research/explore-privacy-research/2017/pet_201711/.

[Xu12] Xu, Heng; Teo, Hock-Hai; Tan, Bernard CY; Agarwal, Ritu: Effects of individual self-protection, industry self-regulation, and government regulation on privacy concerns: A study of location-based services. Information Systems Research, 23(4):1342–1363, 2012.

[Zi15] Zimmermann, Christian: A categorization of transparency-enhancing technologies. arXiv preprint arXiv:1507.04914, 2015.

All websites have been last accessed on December 5th, 2017.

A Demographische Daten der Interviewteilnehmer

Tab. 2: Demographische Daten der Interviewteilnehmer

Code	Branche	Unternehmensgröße Mitarbeiter	Unternehmensgröße Umsatz (in €)	Position	♂/♀	Dauer (hh:mm:ss)
A	Briefgesellschaft	1001-5000	50-100 Mio	Mitglied der Geschäftsleitung, Leiter Marketing und Vertrieb	♂	01:20:16
B	Zahlungssystem-Anbieter	51-200	n.a.	Produktionsmanager	♂	00:45:16
C	Energie-Beratung	11-50	1 Mio.	Geschäftsführer	♂	01:18:48
D	Anbieter E-Commerce-Lösungen	51-200	5-10 Mio.	Leiter Produktionsmanagement	♂	00:55:42
E	Anbieter E-Commerce-Lösungen	51-200	5-10 Mio.	Solutions Manager	♂	01:18:48
F	Anbieter E-Commerce-Lösungen	51-200	5-10 Mio.	Berater technische Pre-Sales	♂	00:44:57
G	Telekommunikation	0,1-0,5 Mio.	50-100 Mrd.	Experte Datenschutz-Audits und-Standards	♂	00:58:16
H	Telekommunikation	0,1-0,5 Mio.	50-100 Mrd.	Stellvertretender Leiter Datenschutz-Audits und-Standards	♂	00:58:16
I	Telekommunikation	0,1-0,5 Mio.	50-100 Mrd.	Leiter Datenschutz für Infrastrukturen und Dienstleistungen	♂	01:14:00
J	Beratung Technikfolgenabschätzung IT	1-10	n.a.	Geschäftsführer	♂	01:51:26
K	Finanz-Dienstleister	50001-0,1 Mio.	20-50 Mrd.	Beraterin geschäftlicher Zahlungsverkehr	♀	01:49:48
L	Beratung IT-Management	1-10	n.a.	Geschäftsführer	♂	00:44:17

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 43

Ein Werkzeug zur automatisierten Analyse von Identitätsdaten-Leaks

Timo Malderle[1], Matthias Wübbeling[1,2], Sven Knauer[1,2], Michael Meier[1,2]

Abstract: Schon vor den Leaks von Dienstleistern wie last.fm, Playstation-Network oder Ashley Madison war Identitätsdiebstahl ein relevantes Thema im Bereich IT-Sicherheit. Die deutsche Gesetzgebung fordert zumeist eine Veröffentlichung der Umstände in relevanten Medien. Trotz öffentlicher Bekanntgabe und Präsenz in einschlägigen Medien erreichen relevante Informationen oft nur wenige Betroffene. Durch solche Veröffentlichungen lässt sich der Missbrauch von personenbezogenen und persönlichen Daten durch Kriminelle weder verhindern noch kontrollieren. Individuelle Benachrichtigungen von Betroffenen können die Folgen von Identitätsdiebstahl abschwächen. Dabei sollten die Benachrichtigungen weiterführende Informationen über den Umfang des Leaks beinhalten, welche die Kritikalität der betroffenen Merkmale darstellen und auch über mögliche Maßnahmen informieren. Um eine individuelle Information auf Basis verfügbarer Identitätsdaten-Leaks zu gewährleisten, müssen diese normalisiert und analysiert werden. Aufgrund der großen Menge kursierender Identitätsdatensammlungen ist eine Automatisierung notwendig. Diese Arbeit dokumentiert eine Implementierung zur automatisierten syntaktisch-, semantischen Analyse und Normalisierung relevanter Merkmale öffentlich verfügbarer Identitätsdaten als Vorbereitung zur individuellen Benachrichtigung von Betroffenen.

Keywords: Identitätsdiebstahl; Identitätsdaten-Leaks; Cyber-Crime; Opfer-Warnung; Reaktive Sicherheit

1 Einleitung

Illegal kopierte Sammlungen von Identitätsdaten-Leaks, im Folgenden nur Leaks genannt, kursieren über unterschiedliche Medien in kriminellen Kreisen. Das Internet ist ein beliebter Platz zum Austausch dafür. Betroffene Personen erfahren häufig erst von der Existenz solcher Leaks, wenn deren eigene Identität illegal verwendet wird und es zu einem Schaden kommt. Selbst wenn Mainstream-Medien über solche Leaks berichten, erfahren viele Personen nicht von der eigenen Betroffenheit. Die Warnung dieser Personen ist daher ein Ziel, das Sicherheitsforscher seit geraumer Zeit untersuchen. Dazu wurden Dienste entwickelt, die jedoch auf das Mitwirken der Betroffenen selbst angewiesen sind. Diese müssen dort vorhandene Daten zum Abgleich angeben und erhalten anschließend Informationen über deren Existenz in vorhandenen Leaks. So ein Vorgehen kann für die

[1] Universität Bonn, Institut für Informatik 4, Friedrich-Ebert-Allee 144, 53113 Bonn
{malderle | matthias.wuebbeling | knauer | mm}@cs.uni-bonn.de
[2] Fraunhofer FKIE, Friedrich-Ebert-Allee 144, 53113 Bonn

doi:10.18420/sicherheit2018_03

breite Masse nicht zielführend sein, weshalb proaktive Warnungen durch geeignete Akteure dringend notwendig sind. Für die eindeutige Zuordnung zu einer betroffenen Person müssen Identitätsmerkmale aus vorhandenen Leaks ermittelt werden. Anschließend lassen sich Betroffene möglicherweise sogar über unterschiedliche, ermittelte Kommunikationswege benachrichtigen, um mögliche Schäden abzuwehren oder um die Betroffenen auf die Folgen von Identitätsdiebstahl vorzubereiten.
Die vorliegende Arbeit präsentiert einen Ansatz für die automatisierte syntaktische und semantische Analyse von Leaks und die Identifikation solcher Identitätsmerkmale, die mittelbar oder unmittelbar die Zuordnung zu einer Person erlauben. Dafür werden im folgenden Kapitel 2 zunächst verwandte Arbeiten vorgestellt, die sich der Analyse von Leaks und der Informierung von Betroffenen widmen. Im anschließenden Kapitel 3 wird die Sammlung von öffentlich verfügbaren Leaks im Internet skizziert, die bereits in einer anderen Arbeit umfangreich vorgestellt wurde. Kapitel 4 demonstriert anschließend die praktische Umsetzung der Analyse in einem Werkzeug und evaluiert die verwendeten Mechanismen zur Klassifikation der Identitätsmerkmale anhand großer öffentlicher Leaks. Zum Abschluss fasst Kapitel 5 die Inhalte zusammen und gibt einen Ausblick über weitere Arbeiten die im Rahmen des Projekts umgesetzt werden.

2 Verwandte Arbeiten

Die von einem Identitätsdiebstahl betroffenen Personen haben nur wenige Möglichkeiten, um zeitnah von einem solchen Vorfall zu erfahren. Eine Möglichkeit ist, die eigene Betroffenheit bei einem geeigneten Identitäts-Informationsdienst zu überprüfen. Beispiele für solche Dienste sind *have i been pwned* [Hu17], der *BSI-Sicherheitstest* [Bu17a] oder der *HPI-Leak-Checker* [Ha17]. Diese Dienste ermöglichen die Überprüfung, ob eine E-Mail-Adresse in einem Identitätsdaten-Leak enthalten ist. Dazu übermittelt der Nutzer seine E-Mail-Adresse über ein Eingabefeld auf der Website des Dienstes. Die dem Dienst vorliegenden Leaks werden anschließend auf das Vorhandensein der eingegebenen E-Mail-Adresse überprüft. Bei einem positiven Ergebnis wird der Anwender informiert, dass Teile seiner Daten in einem dem Dienst vorliegenden Leak vorhanden sind. Gegebenenfalls liefert der Dienst dem Nutzer zusätzlich bekannte Hashes oder Passwörter, welche mit der E-Mail-Adresse gemeinsam geleakt wurden.
Problematisch ist hierbei, dass diese Dienste nur die Überprüfung von E-Mail-Adressen anbieten. Benutzernamen, Telefonnummern oder Zahlungsmitteldaten können damit beispielsweise nicht überprüft werden. Für den Nutzer ist sowohl die Aktualität der den Diensten vorliegenden Leaks undurchsichtig, als auch der Inhalt der Leaks.
Für Online-Dienste, wie Social-Media-Dienste, gibt es verschiedene Ansätze, um zu registrieren, dass von ihnen die Identitätsdaten der Nutzer entwendet worden sind. Methoden mit Watermarking [KK12, SER12] oder Honeypots [BD15] kommen dabei in Frage. Diese Ansätze gehören zu den präventiven Maßnahmen. Sollte es zu einem Identitätsdaten-Leak kommen, hat der Online-Dienst die Möglichkeit dies zu erkennen, um geeignete Maßnahmen einzuleiten. Ob ein Online-Dienst solche Schutzmaßnahmen einsetzt, ist für den Benutzer

nicht ersichtlich. Werden diese Maßnahmen gar nicht oder nur unzureichend eingesetzt, können Identitätsdaten unbemerkt abhanden kommen. Die betroffenen Identitätsinhaber wären einer erhöhten Bedrohung ausgesetzt.
Weitere Arbeiten untersuchen die Verbreitung von Leaks [OMS16], deren Auswirkung auf die Privatsphäre durch die Verkettbarkeit von Identitätsdatensätzen [HN17], sowie deren wirtschaftliche Folgen [Bu16]. Darüber hinaus befasst sich eine Arbeit mit der Verarbeitung von Leaks und der Erkennung von Identitätsmerkmalen [Gr16]. Es ist aber nicht erkennbar, wie das dort vorgestellte System arbeitet. Zusätzlich ist unklar, wie mit Problemen umgegangen wird, die aus der vorgeschlagenen Lösung resultieren.

3 Sammlung von Leaks

Für eine umfassende Analyse öffentlich verfügbarer Leaks müssen diese zunächst gefunden und gesammelt werden. Öffentlich verfügbar ist ein Leak, wenn ein Zugriff ohne Zugangsbeschränkung möglich ist. URLs, die einen randomisierten String im Pfad besitzen, stellen keine grundsätzliche Beschränkung des öffentlichen Zugangs dar. Diese Arbeit basiert auf der technischen Umsetzung zur Sammlung von Leaks aus einer vorangegangen Arbeit [MWM18]. Da die weitere Analyse auf den zuvor gesammelten Leaks basiert, wird im nächsten Abschnitt das Konzept von Datensenken im Allgemeinen vorgestellt. Daran anschließend wird die stichprobenartige manuelle Analyse zur Vorbereitung auf eine automatisierte Auswertung dokumentiert.

3.1 Datensenken

Datensenken sind Speicherorte (beliebiger) Daten im Internet, die über eine URL referenzierbar sind. Einige Dienstleister stellen kostenfrei und ohne weitere Beschränkung Speicherplatz für solche Datensenken zur Verfügung. Dieses Angebot nehmen *Hacker* und *Datenhehler* in Anspruch und verteilen darüber unter anderem auch Leaks mit Identitätsdaten. URLs solcher Leaks werden von Datenhehlern oder von Hackern selbst weitergegeben. Dafür nutzen diese weitere Dienste wie spezialisierte Websites, Forensysteme oder andere soziale Medien.
Für die Sammlung von Leaks müssen die URLs der Datensenken verwendet werden, von denen die Inhalte der Leaks geladen werden. Der Erhalt der URLs ist dabei häufig automatisiert möglich, beispielsweise wenn Datensenken eine API mit entsprechenden Funktionen anbieten. Spezielle *Leak-Monitoring-Pages* aggregieren URLs mit unterschiedlich hohem Aufwand, die nicht unmittelbar automatisiert ermittelt werden können. Für die manuelle Untersuchung wurden Leaks sowohl von automatisiert als auch von nicht automatisiert nutzbaren Datensenken geladen. [MWM18]

3.2 Analyse der gesammelten Daten

Für die erste Auswertung der gesammelten Daten wurden 531 Leaks geladen. Zusammen verfügen diese Leaks über eine Sammlung von 3,33 Milliarden E-Mail-Adressen, wobei es nach Abzug der Duplikate noch 1,56 Milliarden E-Mail-Adressen sind. Vergleichbar ist die gesammelte Menge an E-Mail-Adressen mit der Menge an E-Mail-Adressen verwandter Dienste wie *have i been pwned* (4,72 Milliarden), wobei hier auch E-Mail-Listen (sogenannte Spam-Listen) ohne sonstige Informationen wie Passwörter verwendet wurden [Hu17]. Ein anderer Dienst namens *Vigilante.pw* besitzt ca. 3,56 Milliarden E-Mail-Adressen [vi17]. Die Größe einzelner Leaks variiert stark. Ebenso variabel gestaltet sich das Format der Daten. Im Laufe der Analyse wurde festgestellt, dass es keinen De-facto-Standard für das Format von Leaks gibt, sondern sich jeder Angreifer oder Datenhehler eines mehr oder weniger eigenen Formates bedient. Der Großteil der Leaks (394 von 533) wurde als CSV/TXT Datei veröffentlicht, gefolgt von (teils unvollständigen) SQL-Dateien (126). Aber auch andere Formate wie PHP (8), JSON (3), HTML (1) und XLS (1) waren vertreten. Zudem sind die Leaks häufig mit weiteren Daten wie ASCII-Art oder ähnlichen Szene-Merkmalen versehen. Da der Formatwechsel oftmals auch innerhalb einer Datei stattfindet, kann von einer Aggregation der Inhalte ausgegangen werden. Diese inhomogene Strukturierung der Daten ist für eine generische Syntaxanalyse ein mögliches Hindernis. Als Reaktion wird während der automatisierten Analyse jeder Datensatz aus einem Leak entsprechend seiner zeilenweisen Struktur in Blöcke unterteilt und anschließend weiter mittels Parser analysiert. Hierbei wird festgestellt, dass die Datensätze in den geleakten Dateien zumeist eine der folgenden Formen besitzen, welche in Teilen oder in einem gesamten Leak genutzt werden:

```
e-mail-adr.:passwort
e-mail-adr.:passwort-hash
e-mail-adr.:passwort-hash:klartext-passwort
e-mail-adr.:benutzername:paswort-hash:salt
nutzer-id:benutzername:e-mail-adr.:ip-adr:passwort-hash:salt
benutzername:e-mail-adr.:passwort-hash:salt:geburstag:klartext-passwort
Gesamte SQL-Tabelle als txt/csv
Ein kompletter SQL-Dump
```

List. 1: Beispiele möglicher Anordnungen von Identitätsmerkmalen in einem Leak

In List. 1 ist zu sehen, dass Trennzeichen dazu genutzt werden, um die in einem Datensatz vorliegenden Identitätsmerkmale voneinander zu trennen. Statt des in der vorherigen Auflistung genutzten Trennzeichens *Doppelpunkt* werden auch weitere Trennzeichen wie *Semikolon*, *Komma*, *Tab*, '\t', '\r', *Leerzeichen* oder ähnliche genutzt. Da nicht alle möglichen Trennzeichen zuvor bekannt sind, wird ein dynamischer Ansatz für die Erkennung genutzt. Dabei wird die syntaktische Analyse zusammen mit der semantischen Analyse durchgeführt.

Neben der Analyse der manuell gesammelten Daten wurden auch die automatisiert gesammelten Daten untersucht. Diese stammen aus einer Untergruppe der Datensenken, genannt Paste-Pages. Paste-Pages bezeichnen Websites, welche es ermöglichen, text-basierte Inhalte ohne Inhaltskontrolle öffentlich verfügbar zu verbreiten. Hierfür platziert der Nutzer seine

Inhalte auf der Website, wie beispielsweise Ausschnitte aus einer geleakten Datenbank, und teilt den generierten Link sichtbar für mögliche Interessenten. Die selektierten Paste-Pages lieferten täglich im arithmetischen Mittel 91 neue, relevante, aber nicht duplikatfreie Pastes für die Datenbank. Es konnten von April 2017 bis November 2017 insgesamt 16.000 Pastes gesammelt werden, die zusammen 14.168.206 Millionen E-Mail-Adressen enthalten.

4 Werkzeug zur automatisierten Analyse

Zur effizienten Benachrichtigung von Betroffenen nach einem Identitätsdatendiebstahl wird ein Werkzeug benötigt, welches gesammelte Leaks automatisch analysiert. Das Werkzeug soll unterschiedliche Detailtiefen bei der Analyse unterstützen. Beispielsweise kann ein Leak ausschließlich mit einem regulären Ausdruck nach E-Mail-Adressen durchsucht werden. Allerdings gehen zusätzliche Informationen aus dem Leak verloren. Der genannte oberflächliche Ansatz ist deswegen nicht zielführend. Ein erweiterter Ansatz ist, die Daten so zu analysieren, dass möglichst viele Identitätsmerkmale erkannt werden. Als Identitätsmerkmale werden alle Attribute einer Identität bezeichnet, welche dieser individuell zugerechnet werden können. Dies können neben persönlichen Merkmalen wie Name oder Geburtsdatum auch Identifikatoren wie Email-Adressen, Benutzernamen und zugehörige Passwörter sein. Ebenfalls zu den Identitätsmerkmalen zählen Attribute wie IBAN, Kreditkartennummern, Adresse oder Telefonnummern. Anhand der Analyse dieser Merkmale kann ein genaueres Risiko für diese Identität erhoben werden. Eine effektive und zeitnahe Benachrichtigung eines Betroffenen ist mit höherer Dringlichkeit zu betrachten, wenn Konto- oder Kreditkartendaten veröffentlicht wurden, als wenn Zugangsdaten eines Browsergames betroffen sind. Die detaillierte Analyse eines Leaks hat den Vorteil, dass sich weitere Kommunikationskanäle zur Warnung ergeben. Eine Anschrift etwa kann für eine postalische Benachrichtigung verwendet werden. Um dies umzusetzen, müssen Identitätsdaten syntaktisch geordnet werden, um sie anschließend semantisch zu analysieren. Dazu werden zunächst Trennzeichen erkannt, um anschließend die Identitätsmerkmale genauer zu untersuchen.

4.1 Erkennung der Trennzeichen

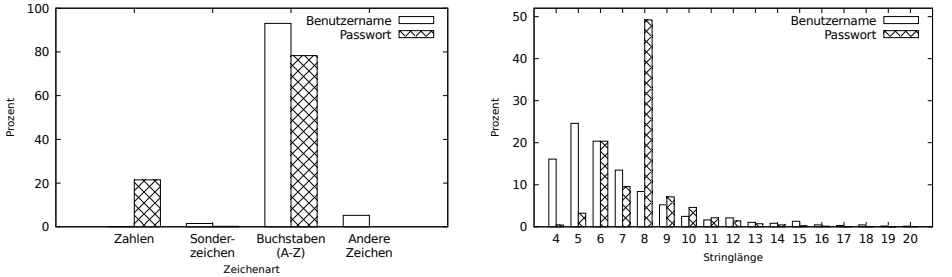
Die Identitätsmerkmale innerhalb eines Leaks sind mit einem Trennzeichen voneinander abgetrennt. Dabei können grundsätzlich beliebige Trennzeichen verwendet werden. Eine Möglichkeit zur automatisierten Analyse ist, die gängigsten Trennzeichen durch eine manuelle Analyse zu ermitteln. Wird in einem Leak allerdings ein anderes Trennzeichen verwendet, so können die Merkmale nicht mehr mittels Parser analysiert werden. Eine dynamische Erkennung der Trennzeichen ist demnach zielführender. Dazu werden zunächst reguläre Ausdrücke genutzt, die auch später zur Merkmalserkennung verwendet werden. Diese werden zeilenweise auf die Leaks angewandt, um die Zeichen vor und nach dem erkannten Ausdruck zu ermitteln. Aufeinander folgende Zeilen mit denselben syntaktischen

Strukturen werden über die Häufigkeit der ermittelten Trennzeichen zugeordnet. Alle zusammenhängenden Zeilen mit derselben Trennzeichen-Syntax werden zu einem Block zusammengefasst. Wird für eine Zeile ein anderes Trennzeichen als bei den vorangegangenen Zeilen ermittelt, beginnt das Verfahren für den neuen Block von vorne.

4.2 Erkennung von Identitätsmerkmalen

Für den Entwurf eines solchen Systems ist eine Identifizierung der Identitätsmerkmale notwendig, welche durch einen Parser automatisiert erkannt werden können. Durch eine manuelle Analyse der Daten kann festgestellt werden, dass folgende Merkmale in den Leaks vorkommen: *E-Mail-Adresse, Passwort-Hash, Salt, Passwort, Benutzername, Vorname, Nachname, Anschrift, Telefonnummer, IBAN, Kontonummer, Bankleitzahl, Kreditkartennummer, Geburtsdatum, ...* . Diese Merkmale besitzen verschiedene Eigenschaften. Beispielsweise lassen sich, wie schon zuvor gesagt, E-Mail-Adressen zuverlässig über die Syntax erkennen, da die Syntax vollständig definiert ist [Re01]. Kreditkartennummern besitzen auch eine erkennbare Syntax, allerdings existieren auch andere Zahlen von gleicher Länge, die als *False-Positive* erkannt werden. Kreditkartennummern besitzen eine Länge von 12 bis 19 Stellen und an der letzten Stelle eine Prüfnummer [Am16]. Um einen Großteil der *False-Positives* auszusortieren, kann bei jedem Wert, der über die Länge erkannt wurde, die Prüfnummer berechnet und kontrolliert werden. Eine effektive Erkennung der einzelnen Identitätsmerkmale anhand der Syntax ist abhängig von dem jeweiligen Identitätsmerkmal. Problematisch ist die Erkennung allerdings bei den Merkmalen *Passwort, Benutzername, Vorname, Name*. Ein Vor- und Nachname sollte in der Regel nur aus Buchstaben bestehen. Ob diese Einschränkung von dem jeweiligen Online-Dienst umgesetzt wurde ist offen. Generell bestehen diese vier Merkmale aus einem beliebig langen String, wobei dieser unter Umständen aus Buchstaben, Zahlen und Sonderzeichen bestehen darf. Eine Unterscheidung über die Syntax ist demnach nicht trivial.
Es wird ein weiterer Ansatz zur Erkennung der Merkmale benötigt. Dazu werden Listen recherchiert, welche die am häufigsten vorkommenden Elemente eines Merkmals beinhalten. Diese Listen werden mit den Elementen einer Spalte verglichen [Gr16]. Es werden Listen zur Erkennung der folgenden Merkmale genutzt: Passwort, Vorname [Bu17b, IA16], Nachname [Bu17b, Ve17], Bankleitzahl (vollständige Liste deutscher Banken) [De17]. Diese Listen bilden eine Sammlung von Vergleichslisten $L_1, ..., L_n$. Jede Liste enthält v Vergleichsbegriffe: $L_{1...n} = \{l \mid l \in [l_1, ..., l_v]\}$. Die zu analysierende Spalte M besitzt w Elemente: $M = \{m \mid m \in [m_1, ..., m_w]\}$. Gesucht ist die Liste L_t, die die größte Schnittmenge mit der gegebenen Spalte besitzt: $L_t = \{L_i \mid (|L_i \cap M|) \geq (|L_j \cap M|) \mid \forall i, j \in [1, ..., n]\}$. Im Anschluss muss ein Schwellwert y ermittelt werden, der von $(|L_t \cap M|)$ überschritten werden muss, um ein Ergebnis aufgrund des Grundrauschens auszuschließen: $\frac{(|L_t \cap M|)}{|M|} > y$. Zusammengefasst wird für eine bestimmte Spalte die Liste gesucht, die mit ihren Elementen am besten mit der Spalte übereinstimmt. Sollte die Übereinstimmung groß genug sein, so kann auf diese Weise auf den Inhalt der Spalte geschlossen werden.
Problematisch ist allerdings die Erkennung der Benutzernamen, da hierfür keine für das

Projekt ausreichende Liste gefunden werden kann. Aufgrund dessen wird ein weiteres Vorgehen benötigt. Eine weitere Möglichkeit zur Unterscheidung von Passwort-Spalten und Benutzernamen-Spalten ist es, dies anhand der Verteilung von Stringlängen und Zeichen der Merkmale aus den jeweiligen Spalten durchzuführen. Zu Untersuchung dieses Vorgehens wird der Leak *badoo* verwendet, da dieser Leak sowohl Klartextpasswörter als auch Benutzernamen enthält. Der Leak *badoo* enthält 112 Millionen Identitätsdatensätze. Teil eines solchen Datensatzes ist der Passwort-Hash, der Benutzername, die Telefonnummer und abhängig von der Version des Leaks auch das geknackte Passwort. Problematisch ist bei der Erkennung der Spalten bei diesem Leak lediglich die Passwort- und Benutzernamen-Spalte, da hierzu auf den ersten Blick ein semantisches Verständnis benötigt wird. Der Passwort-Hash, die E-Mail-Adresse, als auch die Telefonnummern können über die Syntax erkannt werden.
Vorstellbar ist, dass auch das Passwort und der Benutzername auf einer syntaktischen Ebene unterschieden werden können, indem die Verteilung von Passwortlängen und der Passwort-Buchstaben analysiert wird. Dies wird nun genauer betrachtet. In Abbildung 1 werden die Ergebnisse einer syntaktischen Untersuchung dargestellt. Es wurden aus dem genannten Leak die Spalten mit Benutzernamen und Passwörtern analysiert, indem die unterschiedlich oft vorkommenden Zeichen gezählt und die Längen der jeweiligen Merkmale festgehalten werden. Die Abbildung 1a zeigt die prozentuale Verteilung von Zeichen der jeweiligen Spalte in die Kategorien *Zahlen*, *Sonderzeichen*, *Buchstaben (A-Z)* und *andere*. Es ist zu erkennen, dass Passwörter einen deutlich höheren Anteil an Zahlen als Benutzernamen besitzen. In der Abbildung 1b ist die Häufigkeit der String-Längen von Passwörtern und Benutzernamen dargestellt. Auffällig ist, dass Passwörter mit der Zeichenlänge von acht Zeichen deutlich häufiger vorkommen, als Benutzernamen mit acht Zeichen. Diese Verteilungen können dazu genutzt werden, um Benutzernamen- von Passwort-Spalten zu unterscheiden.



(a) Prozentuale Verteilung von Zeichenarten

(b) Prozentuale Verteilung der Stringlänge

Abb. 1: Eigenschaften von Passwort und Benutzername

4.3 Anforderungen und Umsetzung

Um in allen vorliegenden Leaks die im vorigen Abschnitt beschriebenen Identitätsmerkmale möglichst eindeutig zu identifizieren, wird ein Framework für die Normalisierung der Daten erstellt. Ausgehend von einem einzelnen Leak zeigt Abbildung 2 die beteiligten Module und den Datenfluss der Normalisierung.
Das *Input-Modul* importiert Rohdaten von Leaks in unterschiedlichen Formaten (z.B. Archive, CSV- oder SQL-Dateien) und normalisiert diese, bevor die Daten an den *Separator* weitergereicht werden. Der *Separator* ermittelt die innerhalb des Leaks verwendeten vorhandenen Trennzeichen. Hierzu wird der Leak zeilenweise durchgegangen und per regulärem Ausdruck eindeutig erkennbare Felder, wie bspw. Email-Adressen, detektiert. Das Zeichen links und rechts vom gefundenen Feld wird extrahiert und als mögliches Trennzeichen vermerkt. Anschließend wird das gefundene Zeichen als Trennzeichen für die folgenden Zeilen verwendet und die Anzahl der so getrennten Spalten auf Gleichheit überprüft. So ermittelte Zeilen mit gleichem Trennzeichen und gleicher Spaltenanzahl werden als Block übergeben. Dieses Vorgehen ist notwendig, da viele Leak-Dateien eine Aggregation unterschiedlicher anderer Leaks darstellen. Das führt mitunter dazu, dass innerhalb einer Datei unterschiedliche Trennzeichen und auch eine unterschiedliche Reihenfolge der Spalten vorliegen können. Um diese Möglichkeit zu berücksichtigen, ermittelt der *Separator* möglichst große Blöcke, die dasselbe Trennzeichen verwenden. An diesem Punkt ist davon auszugehen, dass erkannte Blöcke zusammengehören und die Semantik der Spalten über den gesamten Block dieselbe ist.

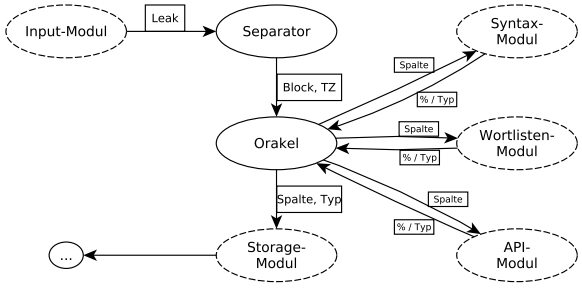


Abb. 2: Beteiligte Module und Datenfluss bei der Normalisierung und Speicherung von Leakdaten

Jeder Block wird mit dem ermittelten Trennzeichen an das *Orakel* übergeben. Das *Orakel* ermittelt die Semantik der Spalten eines Blocks. Dabei verwendet das *Orakel* in der aktuellen Ausprägung drei Module, die bei der Auswahl der Spalten-Semantik unterstützen: das *Syntax-Modul*, das *Wortlisten-Modul* und das *API-Modul*. Das *Orakel* teilt die Blöcke anhand des Trennzeichens in einzelne Spalten und übermittelt diese jeweils an die entwickelten Module. Als Rückgabe erhält das *Orakel* den Anteil der Übereinstimmung mit der Wortliste.

Das *Syntax-Modul* erlaubt die Zuordnung einer Spalte zu Bedeutungen, die anhand von Automaten ermittelt werden können. Dazu gehören vor allem solche Datensätze, die bei der Erstellung bestimmten Regeln folgen, die über reguläre Ausdrücke zu prüfen sind. Das *Syntax-Modul* prüft dabei mit regulären Ausdrücken auf die folgenden Typen: E-Mail, Telefonnummer, Bankleitzahl, Kontonummer, Kreditkartennummer, Hashwerte (z.B. gehashter Passwörter mit Salz, etc.), Datum, IP-Adressen. Dabei werden anschließend auch

Prüfsummen berechnet, wie etwa bei Kreditkartennummern, um nur Typ-konforme Daten zu berücksichtigen.

Das *Wortlisten-Modul* ermöglicht dem *Orakel* den Zugriff auf entsprechende Wortlisten. Dabei soll das gesuchte Merkmal durch einen Vergleich mit spezifischen Wortlisten identifiziert werden. Über das *Wortlisten-Modul* lassen sich die folgenden semantischen Typen von Spalten erkennen: Vornamen, Nachnamen, Klartextpasswörter. Um die Merkmale möglichst genau zu unterscheiden, werden spezifische Listen der entsprechenden Typen verwendet. Am Beispiel von Klartextpasswörtern lässt sich darstellen, weshalb umfangreiche Listen mit vielen Elementen nicht vorteilhaft bei der Merkmalserkennung sind. Passwörter besitzen durch die hohe Individualität bestenfalls eine hohe Entropie. Durch die große Menge individueller Passwörter würde bei der Berücksichtigung aller bekannten Passwörter eine hohe Anzahl an Zeichenkombinationen als Passwort erkannt, da diese auch als Passwort verwendet werden. Daher würde eine umfangreiche Wortliste aus Passwörtern mit hoher Wahrscheinlichkeit bei einer Spalte mit Vornamen positiv spezifizieren, möglicherweise sogar viel deutlicher, als eine Vornamens-Wortliste. Bei einer Beschränkung auf die gängigsten Namen und Passwörter, werden zwar die Trefferraten geringer, dafür werden die Aussagen genauer und einfacher differenzierbar, da viele False-Positives vermieden werden.

In Abbildung 3 ist die Evaluation der Wortlisten zu erkennen. Ausgeführt wird die Evaluation der genutzten Listen und des gedachten Vorgehens auf 100.000 Datensätzen des Leaks der *Modern-Business-Solutions*, welche Identitätsdaten aus der Automobilbranche und Personalvermittlung enthalten [Ch16]. Dieser Leak wird genutzt, da Vornamen und Nachnamen enthalten sind. Zur Evaluation der Passworterkennung werden 100.000 Datensätze eines Leaks einer Dating-Website verwendet, da dieser Leak Klartextpasswörtern beinhaltet [Wh16]. Beide Leaks wurden zunächst in einer manuellen Klassifikation begutachtet und die Spalten für Passwörter, Vornamen und Nachnamen identifiziert, alle anderen Daten wurden nicht weiter berücksichtigt. In der Evaluation wird nun getestet, wie signifikant sich Vornamen-, Nachnamen-, und Passwort-Spalten mit den ermittelten Wortlisten unterscheiden lassen. In Abbildung 3 sind auf der X-Achse drei Kategorien zu sehen: Passwort-, Vornamen-, und Nachnamen-Detektion. Jede dieser Kategorien besitzt eine Spalte mit 100.000 Attributen aus den zuvor genannten Leaks als Eingabe. Die verschiedenen Balken in der Abbildung stehen für unterschiedliche, getestete Wortlisten. Die Höhe eines Balkens gibt an, wie hoch die Übereinstimmung der jeweiligen Spalte aus dem Leak mit der Wortliste ist. Die Abbildung zeigt deutlich, dass die gewählten Listen eine gute Spezifizität besitzen, um die Semantik einer Spalte zu ermitteln. Bei dem Test der Passwort-Detektion ist zu erkennen, dass alle getesteten Passwort-Listen Übereinstimmungen mit der Passwort-Spalte des Leaks besitzen. Die anderen Wortlisten besitzen jedoch keine nennenswerte Übereinstimmung. Ein Passwort lässt sich somit effektiv erkennen. Bei den anderen Detektionen haben die Listen von der gleichen Kategorie wie die zu testenden Spalten auch eine deutlich höhere Übereinstimmung. Auch Vornamen und Nachnamen lassen sich somit effektiv erkennen.

Es wird ebenfalls aus Abbildung 3 deutlich, dass bereits kleine Listen ausreichen, um sich von den anderen Wortlisten-Typen zu unterscheiden. Obwohl die Übereinstimmung der

überprüften Passwort-Listen im einstelligen Bereich liegt, sind die Werte deutlich höher als die Rate der Vor- beziehungsweise Nachnamenslisten. Ebenfalls bemerkenswert ist die Tatsache, dass die Top-5.000 und Top-10.000 Passwort-Listen viele Vornamen enthalten. Die Trefferrate ist sogar deutlich höher als bei den Klartextpasswörtern selbst. Dies stellt an dieser Stelle kein Problem dar, weil die Vornamen-Listen deutlich höhere Trefferraten liefern und die Semantik den Spalten damit gut zugeordnet werden kann. Um die geringen Trefferraten der Passwort-Liste gegen False-Positive abzusichern, wird die im vorigen Abschnitt erwähnte Entropieanalyse im Anschluss durch das *Orakel* durchgeführt.

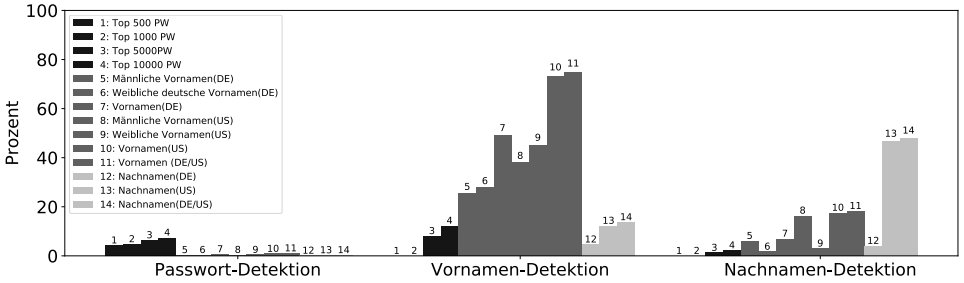


Abb. 3: Detektion von Merkmalen mit dem Wortlistenmodul

Als drittes Modul wird das *API-Modul* genutzt, um die Werte der jeweiligen Spalte gegen existierende Web- oder Offline-APIs zu testen. In der aktuellen Version soll die API eines großen Karten-Dienstleisters verwendet werden, um Postleitzahlen sowie Adressdaten zu erkennen. Auch für Telefonnummern lassen sich entsprechende APIs verwenden, wenn auch nur die Nummern erkannt werden, die im Telefonbuch eingetragen sind.

Nachdem das *Orakel* alle Spalten semantisch zugeordnet hat, werden die Daten der Spalte gemeinsam mit dem semantischen Typ der Spalte an das *Storage*-Modul weitergereicht. Das *Storage*-Modul unterstützt verschiedene Backends zur Speicherung der Leakdaten. Die aktuelle Implementierung speichert die Daten in einer MongoDB-Instanz. Aufgrund der geltenden Datenschutzgesetze sollen an dieser Stelle die Maßgaben der datenschutzrechtlichen Begleitung beachtet und umgesetzt werden. Insbesondere betrifft dies Datenreduktionsverfahren, beispielsweise zum Umsetzen von Sperrlisten auf Basis von Prüfsummen.

5 Zusammenfassung

Veröffentlichungen von Leaks mit Identitätsdaten sind heutzutage fast schon an der Tagesordnung. Die Dunkelziffer existierender Leaks ist vermutlich sehr viel höher. Es wird davon ausgegangen, dass viele Betroffene tatsächlich erst durch einen Schadenseintritt, also den tatsächlichen Identitätsdiebstahl, davon erfahren, dass ihre Identitätsdaten zuvor von Kriminellen kopiert wurden. Eine frühzeitige Benachrichtigung und Warnung der Betroffenen bereits nach dem Kopieren der Identitätsdaten durch Unbefugte ist daher wünschenswert.

Um Leaks zu verkaufen oder um die eigene Reputation zu erhöhen, veröffentlichen Kriminelle regelmäßig vorhandene Leaks ganz oder teilweise. Die weite Verbreitung durch solche Veröffentlichungen erhöht das Missbrauchspotential deutlich. Aus der Verfügbarkeit ergibt sich aber eine Möglichkeit für geeignete Akteure, Betroffene zeitnah zu ermitteln und über Risiken und Vorsichtsmaßnahmen zu informieren. Eine möglichst automatisierte Umsetzung zur Analyse von Leaks und Warnung von Betroffenen ist für die Verarbeitung der großen Datenmengen geboten.
Diese Arbeit demonstriert die automatisierte Ermittlung relevanter Identitätsmerkmale öffentlich verfügbarer Leaks. Dabei ist davon auszugehen, dass der Erhalt der Leaks von entsprechenden Datensenken technisch bereits einsatzbereit ist. Die Sammlungen von Identitätsdaten besitzen meist eine unbekannte und oftmals inhomogene Struktur, sowohl bezogen auf die Syntax als auch auf die Semantik der enthaltenen Daten.
Basierend auf regulären Ausdrücken, Prüfsummen, Wortlisten und öffentlich verfügbaren APIs wurde die Erkennung von Syntax und Semantik an öffentlich verfügbaren Leaks demonstriert. Die Zuverlässigkeit regulärer Ausdrücke wird dabei ergänzt um die Heuristik von Wortlisten und den Zugriff auf große Datensammlungen über API-Anbieter wie Geolocation-Dienste. Die Treffer-Quoten der jeweiligen Wortlisten wurden evaluiert und hinsichtlich falsch-positiver Zuordnungen optimiert.
Insgesamt bieten die in dieser Arbeit vorgestellten Ansätze zur automatisierten Analyse von Identitätsdaten-Leaks die Grundlage für die weitere Identifikation der betroffenen Personen auf Basis der enthaltenen Persönlichkeitsmerkmale. Zur Erreichung des intendierten Ziels einer effektiven Warnung Betroffener verbleiben jedoch offene Herausforderungen für künftige Arbeiten. Hierzu zählt insbesondere die Validierung von Identitätsdaten (Gültigkeit der Zugangsdaten) und im Zusammenhang damit eine Quantifizierung des für Betroffene bestehenden Risikos. Darüber hinaus ist zu untersuchen, welcher Art die Kommunikation und der dazu gewählte Kommunikationskanal zum Betroffenen entsprechen muss, sodass beim Opfer eine angemessene Wahrnehmung erreicht wird. Schließlich bleibt die Frage, wem die Anwendung der vorgeschlagenen Vorgehensweise zur Warnung Betroffener obliegen soll; fällt dies in den Aufgabenbereich öffentlicher Einrichtungen, gehört es zum Verantwortungsbereich der Dienstbetreiber oder sind zusätzliche Akteure gefordert.

Die Autoren danken dem Bundesamt für Bildung und Forschung (BMBF) für die Förderung des Projekts *EIDI* unter dem Förderkennzeichen 16KIS0696K.

Literaturverzeichnis

[Am16] American National Standards Institute: Announcing Major Changes to the Issuer Identification Number (IIN) Standard. `https://www.ansi.org/news_publications/news_story?articleid=da7bcb04-0654-4e03-af54-0e55d50b93a8`, 2016. Sichtung: 11.12.2017.

[BD15] Baykara, M.; Daş, R.: A Survey on Potential applications of Honeypot Technology in Intrusion Detection Systems. International Journal of Computer Networks and Applications (IJCNA), 2 Issue 5:202–211, 2015.

[Bu16] Bundeskriminalamt: Bundeslagebild Cybercrime 2016. https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2016.html, 2016. Sichtung: 11.12.2017.

[Bu17a] Bundesamt für Sicherheit in der Informationstechnik: BSI-Sicherheitstest. https://www.sicherheitstest.bsi.de/, 2017.

[Bu17b] Butler, R.: Most Common First Names and Last Names in the U.S. https://names.mongabay.com/, 2017. Sichtung: 11.12.2017.

[Ch16] Christian Hirsch (Heise Online): Offene Datenbank: 58 Millionen Datensätze im Umlauf. https://www.heise.de/newsticker/meldung/Offene-Datenbank-58-Millionen-Datensaetze-im-Umlauf-3351104.html, 2016. Sichtung: 11.12.2017.

[De17] Deutsche Bundesbank: Download - Bankleitzahlen. https://www.bundesbank.de/Redaktion/DE/Standardartikel/Aufgaben/Unbarer_Zahlungsverkehr/bankleitzahlen_download.html, 2017. Sichtung: 11.12.2017.

[Gr16] Graupner, H.; Jaeger, D.; Cheng, F.; Meinel, C.: Automated Parsing and Interpretation of Identity Leaks. S. 127–134, 2016.

[Ha17] Hasso-Plattner-Institut für Digital Engineering gGmbH: HPI Leak Checker. https://sec.hpi.de/leak-checker, 2017. Sichtung: 23.08.2017.

[HN17] Heen, O.; Neumann, C.: On the Privacy Impacts of Publicly Leaked Password Databases. In (Polychronakis, M.; Meier, M., Hrsg.): DIMVA 2017, S. 347–365. Springer, C., 2017.

[Hu17] Hunt, T.: have i been pwned? https://haveibeenpwned.com, 2017. Sichtung: 11.12.2017.

[IA16] IA7.de - InternetAgentur: Die schönsten Vornamen. http://www.mybabysitter.de/extras/vornamen/, 2016. Sichtung: 11.12.2017.

[KK12] Kale, S.A.; Kulkarni, S.V.: Data Leakage Detection: A Survey. IOSR Journal of Computer Engineering (IOSRJCE)), 1 Issue 6:32–35, 2012.

[MWM18] Malderle, T.; Wübbeling, M.; Meier, M.: Sammlung geleakter Identitätsdaten zur Vorbereitung proaktiver Opfer-Warnung. In: MKWI 2018. 2018. Wird auf Anfrage zur Verfügung gestellt.

[OMS16] Onaolapo, J.; Mariconti, E.; Stringhini, G.: What Happens After You Are Pwnd: Understanding the Use of Leaked Webmail Credentials in the Wild. IMC '16 Proceedings of the 2016 Internet Measurement Conference, S. 65–79, 2016.

[Re01] Resnick, P.: Internet Message Format. RFC 2822, RFC Editor, April 2001.

[SER12] Shabtai, A.; Elovici, Y.; Rokach, L.: A Survey of Data Leakage Detection and Prevention Solutions, 2012.

[Ve17] Verein für Computergenealogie e.V. : Die 1000 häufigsten Familiennamen in Deutschland. http://wiki-de.genealogy.net/Die_1000_häufigsten_Familiennamen_in_Deutschland, 2017. Sichtung: 11.12.2017.

[vi17] vigilante: vigilante.pw. https://vigilante.pw/, 2017. Sichtung: 11.12.2017.

[Wh16] Whittaker, Z. (ZDNet): A dating site leaked over a million accounts because of shoddy security. http://www.zdnet.com/article/dating-site-leaked-one-million-accounts-because-of-shoddy-security/, 2016. Sichtung: 11.12.2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 55

Hashing of personally identifiable information is not sufficient

Matthias Marx[1], Ephraim Zimmer[1], Tobias Mueller[1], Maximilian Blochberger[1], Hannes Federrath[1]

Abstract: It is common practice of web tracking services to hash personally identifiable information (PII), e. g., e-mail or IP addresses, in order to avoid linkability between collected data sets of web tracking services and the corresponding users while still preserving the ability to update and merge data sets associated to the very same user over time. Consequently, these services argue to be complying with existing privacy laws as the data sets allegedly have been pseudonymised. In this paper, we show that the finite pre-image space of PII is bounded in such a way, that an attack on these hashes is significantly eased both theoretically as well as in practice. As a result, the inference from PII hashes to the corresponding PII is intrinsically faster than by performing a naive brute-force attack. We support this statement by an empirical study of breaking PII hashes in order to show that hashing of PII is not a sufficient pseudonymisation technique.

Keywords: personally identifiable information; hashing; pseudonymisation

1 Introduction

The advertisement industry is driven by the need to deliver ads tailored to the audience. In order to deliver targeted ads on the users' screens, they are being tracked across the web by different companies who often also offer reconciliation services for data to increase the accuracy and the value of their data sets.

Traditionally, users were tracked with web cookies. Cookies, however, are not as reliable as other tracking methods such as browser fingerprinting [Eck10] or simply using a more stable reoccurring token to identify a user, e. g., e-mail address or telephone number, as it imposes much more work for the user to change those personal identifiers. Even MAC and IP addresses can be linked to individual users and thus can be used to track individuals [Inf16; Cun14; ME12]. With these methods, tracking companies can merge their data by matching on the token. Exposing the actual token as plain text, however, carries regulatory and economical risks. For example, exposing the user's plain e-mail address to a competitor allows them to use the e-mail address for other purposes, such as sending targeted advertisements. Valid e-mail addresses are believed to be worth around £85 [Jen12]. Furthermore, several national as well as international privacy laws and regulations, e. g., the

[1] Universität Hamburg, Sicherheit in verteilten Systemen, Vogt-Kölln-Straße 30, 22527 Hamburg, Deutschland
<firstname>.<lastname>@informatik.uni-hamburg.de

doi:10.18420/sicherheit2018_04

European General Data Protection Regulation (GDPR) [Eur16], prohibit trading or selling data which allows to identify a user without their consent. Lastly, losing plain personally identifiable information (PII) due to a data breach incurs a much higher risk of losing reputation than losing non-userlinkable data.

For these reasons, i. e., to track users in a privacy-friendly manner and protect assets, while still preserving the linkability between different data sets, companies like Google [Goo17], Facebook [Fac17], or Oracle [Ora18] try to pseudonymise personally identifiable information (PII) with an efficient cryptographic hash function. This is very often incorrectly labelled as anonymisation instead of pseudonymisation. The GDPR defines pseudonymisation as "the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person" [Eur16, §4.5]. From a technical point of view, a value can be considered a pseudonym if it is not feasible to infer the actual PII which was used to create the pseudonym.

Using cryptographic hash functions seems like a good choice to achieve pseudonymity of PII in that sense. Hash functions operate one-way, so that the resulting hash value cannot be used to calculate the input (pre-image resistance). In addition, cryptographic hash functions are collision resistant, so that it is most likely to return different hash values for different input values. Since hashes are deterministic an attacker can guess the input and then check if the resulting output is equal to a given hash value. Our hypothesis is, that PII is of low entropy which limits the pre-image space of a hash function in such a way, that it is feasible to revert the pseudonymisation even without the use of additional information as per the definition above. We show this for all possible IPv4 and MAC addresses, phone numbers, and a data set of leaked e-mail addresses. Our results indicate that even consumer-grade hardware is sufficient for revealing actual PII for the allegedly pseudonymised values.

The remainder of this paper is structured as follows. We first present other work in the area of hashing PII in Sect. 2. We then provide some background in Sect. 3 and share our data set, environment, and experimental set-up in Sect. 4. Our results are provided in Sect. 5 and discussed in Sect. 6. We finally summarise the results in Sect. 7.

2 Related Work

Narayanan and Shmatikov [NS05] and Bonneau [Bon12] argue, that structured and memorable inputs for hash functions, such as passwords or, especially relevant for this work, e-mail addresses, provide low entropy and are vulnerable to "smart dictionary" attacks. Felten [Fel12] shows that SHA-1 hashes of social security numbers (SSNs) do not provide sufficient protection. Demir et al. [Dem+17] show that attacks on hashed e-mail and MAC addresses are theoretically feasible. They perform the reconstruction of e-mail addresses from a small data set of MD5 hashes and calculate the required time to brute-force all

available MAC addresses by the benchmark results of `hashcat`. Kumar et al. [Kum+07] show that hashing of search tokens is insufficient for protecting search query logs. Tokens often contain sensitive information, such as names, that can be guessed by dictionary attacks. A data set released by AOL in 2006 was used to show that sensitive information could be derived from hashed search tokens in practice.

These related publications show, that the hypotheses about the low entropy of PII and its insufficiency serving as input of hash functions due to its limited pre-image space has been studied several times and is considered as well known. However, we extend that work by considering the most used hash functions in practise, which are believed to provide an adequate protection level of PII, on a large data set of e-mail addresses and show that it is not only theoretically, but also practically feasible to reconstruct a large number of e-mail addresses. We additionally show that it is practically feasible to revert hashes for IPv4 and currently assigned MAC addresses within minutes as well as phone numbers in appropriate time on consumer-grade hardware.

The practical analysis of our work is improved by the observation of Tatli [Tat15], that the process of generating hashes can greatly be sped up by using patterns instead of relying on wordlists or dictionaries. Additionally, the results of inverting e-mail hashes are based on the work of Polakis et al. [Pol+10] and Kumar et al. [Kum+07], who show that valid e-mail addresses can be harvested from names collected in social networking platforms such as Facebook or Twitter as well as from census data.

3 Background

IPv4 and MAC addresses, telephone numbers, and e-mail addresses represent the most reliable PII, that tracking and advertising services can use to identify individual records of their data sets, update a specific record with newly collected data about an individual, and merge data sets traded with other tracking services. The data management platform BlueKai by Oracle, for example, calls these identifiers of users “match keys” and explicitly states to use MD5 and SHA-256 hashed e-mail addresses or phone numbers as well as user’s IP addresses collected from IP headers [Ora18]. Similarly, the organisation infsoft GmbH, which is specialised in indoor-tracking, uses SHA-256 hashed MAC addresses to recognise individuals [inf18]. For this reason, our experiments focus on these four kinds of PII whose properties will be explained in this section.

Internet Protocol version 4 (IPv4) is specified in RFC 791 [Pos81]. It uses 32-bit addresses of fixed length and thus $2^{32} \approx 4.3 \cdot 10^9$ addresses exist. About 600 million addresses are reserved for special purposes, which means that nearly $3.7 \cdot 10^9$ different IPv4 addresses are worth considering when breaking hashes.[2]

[2] See RFCs 990, 1112, 1700, 1918, 2544, 3068, 3927, 5736, 5737, 6598, and 6890

Media Access Control (MAC) addresses are globally unique network interface identifiers. MAC addresses have 48 bit, allowing $2^{48} \approx 2.8 \cdot 10^{14}$ different addresses. The first 24 bit are called "Organisationally Unique Identifier" (OUI) and are controlled by the IEEE, the latter 24 bit are controlled by manufacturers. IEEE publishes a list of assigned OUIs [IEE+06], excluding secret OUIs. The list contains 24 215 entries, which leads to $2^{24} \cdot 24\,215 \approx 4.1 \cdot 10^{11}$ possible MAC addresses.

Telephone Numbers or more specifically their structure is specified by the International Telecommunication Union (ITU) [Int10]. It can be up to 15 digits and consists of a leading country code of 1-3 digits followed by the so called national significant number (NSN), which in turn consists of an area code and the subscriber number. A list of existing country codes is published by the ITU [Int16] and consists of 217 assigned unique entries at the time of writing. So the remaining national significant number of a phone number can only have a length of 14 digits in case of a one digit country code, 13 digits in case of a two digits country code, and 12 digits in case of a three digit country code, resulting in a total number of $2 \cdot 10^{14} + 44 \cdot 10^{13} + 171 \cdot 10^{12} \approx 8.11 \cdot 10^{14}$. This number can further be reduced significantly when additional information about the country code of a phone number is known. Then, the possible area codes of the country, i. e., the digits directly following the country code, and the possible length of the remaining digits, which often are limited by national numbering plans, can be taken into account.[3] Several lists of area codes and corresponding minimal and maximal lengths of the national significant number can be found at [Int17].

E-mail Addresses have a lot more entropy than the previous examples. The general structure of a global e-mail address is `local-part@domain` while the `local-part` can consist of up to 64 ASCII characters [Kle08; Res08]. Furthermore, the case sensitivity of the local-parts is discouraged [Kle08]. We can hence estimate the number of possible `local-parts` limited to loweralpha, numeric, and 20 special characters[4] to be: $(26 + 10 + 20)^{64} = 56^{64} \approx 2^{372} \approx 7.7 \cdot 10^{111}$. There is a maximum of 255 characters in the `domain` part. Additionally, RFC 2821 limits the length of e-mail addresses to 254 characters [Kle01]. The total number of possible e-mail addresses is approx. $2.2 \cdot 10^{120}$, considering the number of 330 million registered domains [Ver17].

4 Methodology

This section describes the environment, that has been used for the evaluation of the resources required to de-pseudonymise hash values that are meant to protect PII. First, the software and hardware setting for our evaluation, second the acquisition of hash values, and third the different calculations that have been performed will be explained.

[3] For example, the German area code 261 is limited to a minimal NSN of 6 digits and to a maximal NSN of 11 digits. Thus only $10^3 + 10^4 + 10^5 + 10^6 + 10^7 + 10^8$ phone numbers of the form +49 261 dd. . .d are possible.

[4] `!#$%&'*+-/=?^_`{|}~.` as recommended by the W3C [W3C17]

4.1 Environment

To de-pseudonymise hashes, we used `hashcat` v4.0.1 [Ste09], a GPU-based password recovery tool, on a desktop computer equipped with an Intel Core i5-6500 CPU at 3.2 GHz, 16 GB RAM, and a Nvidia GeForce GTX 1050 Ti graphics card with 4 GB GDDR5, running an Ubuntu 16.04. The high performance workload profile and optimised kernels for `hashcat` were enabled. The built-in benchmark of `hashcat` reports this setup to be capable of calculating $6.021 \cdot 10^9$ MD5 and $0.844 \cdot 10^9$ SHA-256 hashes per second.

4.2 Acquiring Hashes

For IP and MAC addresses as well as for telephone numbers, one million distinct addresses or numbers have randomly been generated and the hash values of the generated values have been calculated with MD5 and SHA-256.

As randomly generated e-mail addresses would not necessarily look like real e-mail addresses, real e-mail addresses have been acquired in a different way. A list of 700 million leaked credentials (e-mail address and password combinations) [Tro17] has been downloaded. This list has been cleaned up by dismissing the passwords, validating all addresses according to W3C [W3C17] and choosing one million different `.de`-addresses at random. Of these one million random e-mail addresses the MD5 and SHA-265 hash values have been calculated.

4.3 Breaking Hashes

The `hashcat` utility provides several attack types, including combinator, mask, hybrid, and rule-based attacks. Combinator attacks are two-dictionary attacks, where each word from the one dictionary is paired with each word from the second one. Mask attacks are brute-force attacks, where the search space is limited to certain word structures and character sets. A hybrid attack couples combinator and brute-force or mask attack. Rule-based attacks are dictionary attacks where each word serves as password candidate generator. The rule engine of `hashcat` could for example reverse or duplicate words or could prepend, swap, and replace characters. Multiple rules can be combined.

IPv4 Addresses For breaking the IPv4 address hashes, a combinator attack is most appropriate. Two dictionaries have been created, one for all possible left halves and another one for all possible right halves. Each dictionary has $65\,536 = 2^{16}$ entries. Both dictionaries combined result in all 2^{32} possible IPv4 addresses and require about 1 MB of storage. A single dictionary would need about 60 GB of storage. Alternatively, different rules for a mask attack could be set up.

MAC Addresses The first part of a MAC address forms the OUI, which can be found in public databases. Thus a dictionary of OUIs has been built and only the second part needed to be brute-forced, leading to the execution of a hybrid attack.

Telephone Numbers For breaking the telephone number hashes, a mask attack has been executed by acquiring lists of Indonesian, Chinese, and German country and area codes and supplementing these lists with masks depending on the lengths of the possible subscriber numbers. For example, the mask `4930?d?d?d` gives `hashcat` the instruction to probe all numbers from `49 30 000` to `49 30 999`.

E-mail Addresses Numerous attacks have been performed to break the e-mail address hashes, which can be distinguished in the way they probe the local-part of the addresses. For all attacks, lists of the 10 and 100 most common `.de`-domains extracted from an OpenPGP keyserver [PGP18] have been passed to `hashcat`.

First, several mask attacks have been executed. However, they are appropriate only for short local-parts as the runtime of those attack increases exponentially with the length of the provided mask. For one to four characters, the character set of the provided mask included every possible character, which is allowed to occur in the local-part by specification. By increasing the number of characters in subsequent mask attacks, the character set of the provided masks has been reduced.

Second, dictionary attacks with two wordlists have been executed. The first wordlist (from now on referred to as small dictionary) contained $9.7 \cdot 10^6$ first and last names that had been crawled from Facebook [Bow10]. The second wordlist (from now on referred to as large dictionary) contained approx. $3.1 \cdot 10^9$ words from dictionaries, username, and password lists [Hat13].

Lastly, `hashcat`'s rule engine has been utilised and combined with the two wordlists in order to generate several local-part candidates for each word in the wordlists. This included prepending, appending, and the duplication or deletion of single characters, and rotating, reversing, duplicating, and reflecting of words.

5 Results

In this section, the result of our attacks described in the previous section are presented.

5.1 IPv4 Addresses

Theoretical Considerations With the assumption to be able to compute $6 \cdot 10^9$ MD5 hashes (6 Giga hashes) and $844 \cdot 10^6$ SHA-256 hashes (844 Mega hashes) per second, which are the benchmark results of `hashcat` as explained in Sect. 4.1, calculating a MD5 hash

for each address in the whole IPv4 space completes in 0.72 s and calculating the same for SHA-256 hashes would take 5.09 s.[5]

Practical Results Tab. 1 shows the practical results of our combinator attacks on IPv4 address hashes. We de-pseudonymised all MD5 and SHA-256 hashes in under one minute. De-pseudonymisation of SHA-256 hashes took 5 s (20%) longer than de-pseudonymisation of MD5 hashes.

Search Space	Runtime		Recovered Hashes	Recovery Rate
	MD5	SHA-256		
$4.3 \cdot 10^9$	00:00:25	00:00:30	1 000 000	100%

Tab. 1: Practical results of combinator attacks on one million IPv4 address hashes.

5.2 MAC Addresses

Theoretical Considerations The amount of $2^{48} \approx 2.8 \cdot 10^{14}$ different MAC addresses can effectively be reduced by the 24 215 OUIs published by the IEEE (see Sect. 3). Again, assuming the ability to calculate 6 Giga MD5 hashes and 844 Mega SHA-256 hashes per second, the MAC address space can be exhaustively hashed in 1 min 8 s for MD5 and in 8 min 2 s for SHA-256.[6]

Practical Results Tab. 2 shows the practical results of our hybrid attacks on MAC address hashes. All MD5 as well as all SHA-256 hashes could be de-pseudonymised in under 15 min. De-pseudonymisation of SHA-256 hashes took approx. 10 min (three times) longer than de-pseudonymisation of MD5 hashes.

Search Space	Runtime		Recovered Hashes	Recovery Rate
	MD5	SHA-256		
$4.1 \cdot 10^{11}$	00:04:01	00:13:22	1 000 000	100%

Tab. 2: Practical results of hybrid attacks on one million MAC address hashes.

5.3 Telephone Numbers

Theoretical Considerations In order to break a hashed telephone number, one needs to produce at most $2 \cdot 10^{14} + 44 \cdot 10^{13} + 171 \cdot 10^{12} \approx 8.11 \cdot 10^{14}$ hashes (see Sect. 3). At the hashing rate of 6 Giga MD5 hashes and 844 Mega SHA-256 hashes per second, breaking all hashes of any possible number takes at most 1 d 13 h 33 min for MD5 and 11 d 2 h 55 min

[5] $2^{32}/(6 \cdot 10^9) = 0.7158$ and $2^{32}/(844 \cdot 10^6) = 5.0888$

[6] $2^{24} \cdot 24\,215/(6 \cdot 10^9) = 67.71$ and $2^{24} \cdot 24\,215/(844 \cdot 10^6) = 481.351$

for SHA-256.[7] This is the upper limit for breaking any given number. That does not seem to be out of reach for a determined attacker, but it is probably still too slow for a casual de-pseudonymisation attempt. Improvements can be made when additional information about a phone number is available. For example, if we assume the number to be of a certain origin, e. g., either German, Chinese, or Indonesian, we can limit the pre-image space to approx. 10^{12} by considering the possible area codes of these three countries, i. e., the digits directly following the country code, and the possible length of the remaining digits, which often are limited by national numbering plans (see Sect. 3). Thus, all possible German, Indonesian and Chinese telephone numbers can be hashed in 2 min 47 s, assuming a hashing rate of 6 Giga MD5 hashes per second, and can be hashed in 19 min 45 s, assuming a hashing rate of 844 Mega SHA-256 hashes per second.[8] At least theoretically, it is feasible to break the pseudonymity provided by a hashed phone number, especially if we assume previous knowledge about the received pseudonymised data set. This assumption is not far fetched, given that companies exchange data for reconciliation purposes rather than for acquiring new data subjects.

Practical Results Tab. 3 shows the practical results of our mask attacks on telephone number hashes. All MD5 and SHA-256 hashes could be de-pseudonymised. The search space of German and Indonesian telephone numbers is one magnitude larger than the Chinese search space. The runtime difference between de-pseudomisation of MD5 and SHA-256 hashes is below 7 min for a search space of size 10^{11}.

Country	Search Space	Runtime		Recovered Hashes	Recovery Rate
		MD5	SHA-256		
China	$2.3 \cdot 10^{10}$	00:07:17	00:07:12	1 000 000	100%
Germany	$4 \cdot 10^{11}$	02:28:24	02:34:16	1 000 000	100%
Indonesia	$5.8 \cdot 10^{11}$	02:45:57	02:52:42	1 000 000	100%

Tab. 3: Practical results of mask attacks on one million telephone number hashes for three selected countries.

5.4 E-mail Addresses

Theoretical Considerations The large estimated theoretical space for e-mail addresses limited to alpha-numeric and 3 special characters is already unfeasible for performing a brute-force attack, even when ignoring the `domain` part of the e-mail address (see Sect. 3). When only considering the 100 most popular e-mail domains, a total number of $7.88 \cdot 10^{117}$ addresses still exist. In the case of MD5, breaking all hashes would take unimaginable $4.16 \cdot 10^{100}$ years when assuming a hashing rate of 6 Giga hashes per second.[9]

[7] $(8.11 \cdot 10^{14})/(6 \cdot 10^{9}) = 135166.6667$ and $(8.11 \cdot 10^{14})/(844 \cdot 10^{6}) = 960900.4739$

[8] $10^{12}/(6 \cdot 10^{9}) = 166.6667$ and $10^{12}/(844 \cdot 10^{6}) = 1184.8341$

[9] $7.88 \cdot 10^{117}/(6 \cdot 10^{9})/3600/24/365 \approx 4.1646 \cdot 10^{100}$

At first sight, the cost of de-pseudonymising hashed e-mail addresses seems to be overwhelming. But that is assuming an e-mail address consisting of a randomly generated 64 character local-part rather than following a certain structure, e. g., <firstname>.<lastname>@<domain>, <firstname><year>@<domain>, <first letter of firstname>.<lastname>@<domain>, or <7 lowercase letters>@<domain>. Taking a list of the 100 most popular e-mail domains as well as a list of the 1000 most popular first names and the 1000 most popular last names, the four example e-mail structures result in an approx. number of $8 \cdot 10^{11}$ e-mail addresses.[10] With a hashing rate of 6 Giga MD5 hashes per second, we can revert any MD5 hash in 2 min 14 s. With a hashing rate of 844 Mega SHA-256 hashes it would take 15 min 48 s.[11]

Practical Results Tab. 4 gives an overview of the practical results of our mask and dictionary attacks on e-mail address hashes. We observe major differences in runtime depending on attack type, hash algorithm and number of probed domains. For both hashing functions and a combination of all attacks, probing for the top 100 domains takes approx. six times longer than probing for the top 10 domains and results in an increase of 7% of the recovery rate. In total, we de-pseudonymised approx. 43% of the hashed e-mail addresses.

For mask attacks, probing for the top 100 domains takes approx. eight times longer than probing for the top 10 domains and results in an increase of 3.3% of the recovery rate. Computing SHA-256 hashes takes approx. three times longer than computing MD5 hashes.

For small dictionary attacks, probing for the top 100 domains takes approx. six and a half times longer than probing for the top 10 domains and results in an increase of 4.9% of the recovery rate. Computing SHA-256 hashes takes approx. three times longer than computing MD5 hashes.

For large dictionary attacks, probing for the top 100 domains takes approx. five to six times longer than probing for the top 10 domains and results in an increase of 5.6% of the recovery rate. Computing SHA-256 hashes takes approx. two times longer than computing MD5 hashes.

Tab. 5 shows details of the attacks that were conducted using the top 100 domains. The custom charset `?1` consists of loweralpha (`?l`), numeric (`?d`) and special (`-_.`) characters.

6 Discussion

This section analyses the results obtained in the previous section and sets them into the broader context of PII pseudonymisation.

[10] $10^3 \cdot 10^3 \cdot 100 + 10^3 \cdot 10^4 \cdot 100 + 26 \cdot 10^3 \cdot 100 + 26^7 \cdot 100 \approx 8.0428 \cdot 10^{11}$

[11] $(8 \cdot 10^{11})/(6 \cdot 10^9) = 133.3334$ and $(8 \cdot 10^{11})/(844 \cdot 10^6) = 947.8673$

Attack Type	Search Space	Domains	Runtime		Recovered Hashes	Recovery Rate
			MD5	SHA-256		
Mask	$3.5 \cdot 10^{12}$	Top 10	00:38:34	01:57:03	166 152	16.62 %
	$3.5 \cdot 10^{13}$	Top 100	05:18:20	16:40:17	199 377	19.94 %
Small Dictionary	$1.5 \cdot 10^{12}$	Top 10	01:09:40	01:22:57	246 900	24.69 %
	$1.5 \cdot 10^{13}$	Top 100	07:58:29	09:25:29	295 881	29.59 %
Large Dictionary	$3.1 \cdot 10^{12}$	Top 10	01:35:18	02:37:38	281 714	28.17 %
	$3.1 \cdot 10^{13}$	Top 100	07:40:16	15:34:34	337 333	33.73 %
Total	$8.1 \cdot 10^{12}$	Top 10	03:23:32	06:57:38	355 155	35.52 %
	$8.1 \cdot 10^{13}$	Top 100	20:57:05	41:40:20	425 198	42.52 %

Tab. 4: Overview of the practical results of our mask and dictionary attacks on one million e-mail address hashes.

Mask (Length) / Rule	Search Space	Runtime		Recovered Hashes	Recovery Rate
		MD5	SHA-256		
`?l?d?s` (1)	$5.6 \cdot 10^{3}$	00:00:05	00:00:06	11	0.00 %
`?l?d?s` (2)	$3.1 \cdot 10^{5}$	00:00:04	00:00:05	149	0.01 %
`?l?d?s` (3)	$1.8 \cdot 10^{7}$	00:00:04	00:00:07	1 515	0.15 %
`?l?d?s` (4)	$9.8 \cdot 10^{8}$	00:00:04	00:00:07	5 781	0.58 %
`?l?d?l` (5)	$9.0 \cdot 10^{9}$	00:00:08	00:00:22	18 857	1.89 %
`?l?d?l` (6)	$3.5 \cdot 10^{11}$	00:02:33	00:09:09	49 321	4.93 %
`?l?d?l` (7)	$1.4 \cdot 10^{13}$	02:37:01	07:21:16	80 829	8.08 %
`?l` (8)	$2.1 \cdot 10^{13}$	02:38:21	09:09:05	42 914	4.29 %
Small Dictionary (SD)	$9.7 \cdot 10^{8}$	00:01:36	00:01:28	35 117	3.51 %
SD + Set of Rules	$1.9 \cdot 10^{10}$	00:02:25	00:02:45	46 306	4.63 %
SD + `?l`	$3.8 \cdot 10^{10}$	00:03:41	00:04:01	65 282	6.53 %
`?l` + SD	$3.8 \cdot 10^{10}$	00:02:55	00:03:21	51 586	5.16 %
SD + `?l?l`	$1.5 \cdot 10^{12}$	00:46:06	00:57:11	162 852	16.29 %
`?l?l` + SD	$1.5 \cdot 10^{12}$	00:39:01	00:51:15	99 423	9.94 %
`?l` + SD + `?l`	$1.5 \cdot 10^{12}$	00:40:13	00:52:43	85 221	8.52 %
SD + `?d?d?d`	$9.7 \cdot 10^{11}$	00:30:23	00:35:25	26 230	2.62 %
SD + `?d?d?d?d`	$9.7 \cdot 10^{12}$	05:12:19	05:57:20	39 073	3.91 %
Large Dictionary (LD)	$3.1 \cdot 10^{11}$	00:20:41	00:23:53	174 197	17.42 %
LD + Set of Rules	$6.2 \cdot 10^{12}$	01:49:38	03:42:47	206 804	20.68 %
LD + `?l`	$1.2 \cdot 10^{13}$	02:47:50	05:47:16	182 411	18.24 %
`?l` + LD	$1.2 \cdot 10^{13}$	02:42:07	05:40:38	110 193	11.02 %

Tab. 5: Practical results of mask, dictionary and rule-based attacks on one million e-mail address hashes and the top 100 domain. The custom charset `?1` consists of loweralpha (`?l`), numeric (`?d`) and special (`-_.`) characters.

In the previous sections we have shown that the pseudonymisation of PII by means of simple hashing can be subverted relatively easily. We elaborated on the theoretical pre-image space and were able to reduce it with knowledge of either the structure of the data or the data subjects. While we argue that this knowledge is required properly exchanging hashed PII, we acknowledge that it makes our results less applicable to someone who does not have the required knowledge.

According to our experiments with rather modest consumer-grade hardware it seems entirely feasible to break pseudonymisation of certain PII at a low cost. A determined attacker with specialised hardware may very well be able to break pseudonymisation much quicker. Furthermore, the attacks can be easily parallelised on several computers. At the time of writing, Amazon provides computing infrastructure capable of computing 450 Giga hashes per second for about 25 USD per hour [Mat17].

It is obvious that the actually achieved hash rates are much lower than the theoretically calculated values. This partly is due to the fact that not one but one million hashes have been tried to be calculated at the same time. Another reason is that for some attacks the utilisation of the graphics card was below 100 %. For dictionary attacks, words have to be copied from the host system over PCI-Express to the GPU. The GPU is busy during the copy operation and can not compute hashes which reduces the overall performance. `Hashcat`'s rule system modifies words on the GPU which effectively reduces the copy-buffer overhead [Has17]. This effect can be seen from Tab. 5.

We have observed that the advertisement industry uses cryptographic hash functions because they make it very hard to find a pre-image of a given hash. It is thus believed that hashing PII is a good way of pseudonymisation. However, if assuming a data breach in form of a third-party learning the whole list of hashed PII, the attacker may very well not be interested in reversing one particular hash, but rather as many hashes as possible. This is very similar to protecting passwords which tends to be achieved by the use of slower hash function. In 1999, the bcrypt hashing scheme was presented [PM99]. It makes brute force attacks harder by having a relatively expensive key setup phase. In 2015, the Password Hashing Competition [Wet16] awarded Argon2 [BDK16] for providing a good protection level for passwords. The setup of our experiments reported to be able to compute $4 \cdot 10^3$ bcrypt hashes per second which is six magnitudes slower than the $6 \cdot 10^9$ MD5 hashes our modest hardware can compute (cf. Sect. 4.1). An attack on a list of hashed PII will thus get more expensive when a slower hash algorithm is used. Certain attacks, however, are still feasible. For example, a list of 700 million leaked e-mail addresses could still be hashed within two days[12]. Such pre-computed result could then be stored for later lookup purposes. The data management industry could still be interested in changing their algorithms to a much less efficient algorithm in order to significantly increase an attacker's effort and costs when breaking hashes.

Another way to increase the effort of an attacker is to make the attacker break every single

[12] 700000000/4000/60/60/24 = 2.03

hash rather than a whole list of hashes. This can be achieved by prefixing each entry with a "salt" of a length of the desired protection level, e. g., eight bytes. While this allows the continued use of efficient hashing algorithms like MD5 or SHA-256, it requires a change in the format the data is exchanged and traded, because every single data entry has to carry the salt used. This also prevents data sets to be merged easily unless the two sets have used the same salts for each and every individual record. Data management providers might be relunctant to implement such a scheme due that inability of reconciliating data sets.

7 Conclusion

In this paper, experiments with rather modest hardware have been performed in order to quantify the effort required to thwart hash-based pseudonymisation of PII. The results show, that practical attacks are feasible, which complements theoretical analyses of previous work.

In more detail, the performed experiments show, that PII with relatively small pre-image space, namely IPv4 addresses, MAC addresses, and telephone numbers, are not adequately protected with simple hashing. The pseudonymisation of 43 % of hashed e-mail addresses could also been reverted within less than a day. Our results suggest that a more elaborate way of pseudonymisation is required in order to protect data subjects. In our discussion in Sect. 6, recommendations about mitigating attacks on pseudonymisation are provided. Further, the options to only provide a temporary stop-gap measure is discussed, as computing power and economic efficiency tend to increase.

References

[BDK16] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. "Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications". In: *EuroS&P*. IEEE, 2016, pp. 292–302.

[Bon12] Joseph Bonneau. "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords". In: *IEEE S&P*. IEEE, 2012, pp. 538–552.

[Bow10] Ron Bowes. *Return of the Facebook Snatchers*. 2010. URL: `https://blog.skullsecurity.org/?p=887` (visited on 12/14/2017).

[Cun14] Mathieu Cunche. "I know your MAC address: targeted tracking of individual using Wi-Fi". In: *J. Computer Virology and Hacking Techniques* 10.4 (2014), pp. 219–227.

[Dem+17] Levent Demir et al. "The Pitfalls of Hashing for Privacy". In: *Commun. Surveys Tuts.* (99 2017).

[Eck10] Peter Eckersley. "How Unique Is Your Web Browser?" In: *Privacy Enhancing Technologies*. Vol. 6205. Lecture Notes in Computer Science. Springer, 2010, pp. 1–18.

[Eur16] European Parliament. “Regulation (EU) 2016/679 (General Data Protection Regulation)”. In: *Official Journal of the European Union* L119 (May 2016), pp. 1–88. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC.

[Fac17] Facebook. *Das Hochladen der Kundenliste*. 2017. URL: https://www.facebook.com/business/help/112061095610075 (visited on 12/14/2017).

[Fel12] Ed Felten. *Does Hashing Make Data “Anonymous”?* Federal Trade Commission. Apr. 22, 2012. URL: https://www.ftc.gov/node/605301 (visited on 12/14/2017).

[Goo17] Google. *Integrating CRM Data with Google Analytics to create AdWords Remarketing Audiences*. Jan. 2017. URL: https://developers.google.com/analytics/solutions/crm-integration (visited on 12/14/2017).

[Has17] Hashcat. *FAQ*. 2017. URL: https://hashcat.net/wiki/doku.php?id=frequently_asked_questions (visited on 12/14/2017).

[Hat13] Steven Hatfield. *My wordlist now shared*. 2013. URL: https://wp.me/p2HHRm-F (visited on 12/14/2017).

[IEE+06] IEEE Standard Association et al. *IEEE OUI and company id assignments*. 2006. URL: http://standards.ieee.org/regauth/oui/oui.txt (visited on 12/14/2017).

[Inf16] InfoCuria. *Judgment of the Federal Court of Justice in Germany (Second Chamber) in Case C-582/14: Processing of personal data*. Oct. 2016. URL: http://curia.europa.eu/juris/document/document.jsf?text=&docid=184668&doclang=en (visited on 12/14/2017).

[inf18] infsoft GmbH. *FAQ – Indoor Positioning – Data Protection*. 2018. URL: https://www.infsoft.com/technology/faq (visited on 02/14/2018).

[Int10] International Telecommunication Union. *The international public telecommunication numbering plan*. Nov. 18, 2010. URL: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.164-201011-I!!PDF-E&type=items (visited on 12/14/2017).

[Int16] International Telecommunication Union. *List of Recommendation ITU-T E.164 assigned country codes*. 2016. URL: http://handle.itu.int/11.1002/pub/80eefa4f-en (visited on 12/12/2017).

[Int17] International Telecommunication Union. *List of National Numbering Plans*. 2017. URL: https://www.itu.int/oth/T0202.aspx?lang=en&parent=T0202 (visited on 12/12/2017).

[Jen12] Richard Jenkings. *How much is your email address worth?* The Drum. Apr. 4, 2012. URL: http://www.thedrum.com/opinion/2012/04/04/how-much-your-email-address-worth (visited on 12/11/2017).

[Kle01] John C. Klensin. “Simple Mail Transfer Protocol”. In: *RFC* 2821 (2001).

[Kle08] John C. Klensin. “Simple Mail Transfer Protocol”. In: *RFC* 5321 (2008).

[Kum+07] Ravi Kumar et al. “On anonymizing query logs via token-based hashing”. In: *WWW*. ACM, 2007, pp. 629–638.

[Mat17] Iraklis Mathiopoulos. *Running hashcat v4.0.0 in Amazon’s AWS new p3.16xlarge instance*. Medium. Oct. 28, 2017. URL: https://medium.com/@iraklis/e8fab4541e9b (visited on 12/11/2017).

[ME12] A. B. M. Musa and Jakob Eriksson. “Tracking unmodified smartphones using wi-fi monitors”. In: *SenSys*. ACM, 2012, pp. 281–294.

[NS05] Arvind Narayanan and Vitaly Shmatikov. “Fast dictionary attacks on passwords using time-space tradeoff”. In: *ACM CCS*. ACM, 2005, pp. 364–372.

[Ora18] Oracle Corporation. *BlueKai Platform – Offline match integration*. 2018. URL: https://docs.oracle.com/cloud/latest/marketingcs_gs/OMCDA/IntegratingBlueKaiPlatform/DataIngest/offline_match.html (visited on 02/14/2018).

[PGP18] PGP Public Key Server. *Keydump*. Feb. 2, 2018. URL: http://pgp.key-server.io/sks-dump (visited on 02/02/2018).

[PM99] Niels Provos and David Mazieres. “A Future-Adaptable Password Scheme.” In: *USENIX, FREENIX Track*. 1999, pp. 81–91.

[Pol+10] Iasonas Polakis et al. “Using social networks to harvest email addresses”. In: *WPES*. ACM, 2010, pp. 11–20.

[Pos81] Jon Postel. “Internet Protocol”. In: *RFC* 791 (1981).

[Res08] Peter W. Resnick. “Internet Message Format”. In: *RFC* 5322 (2008).

[Ste09] Jens Steube. *Hashcat*. 2009. URL: https://hashcat.net/ (visited on 12/14/2017).

[Tat15] Emin Islam Tatli. “Cracking More Password Hashes With Patterns”. In: *IEEE Trans. Inf. Forensics Security* 10.8 (2015), pp. 1656–1665.

[Tro17] Troy Hunt. *Password reuse, credential stuffing and another billion records in Have I been pwned*. May 5, 2017. URL: https://www.troyhunt.com/password-reuse-credential-stuffing-and-another-1-billion-records-in-have-i-been-pwned/ (visited on 12/14/2017).

[Ver17] VeriSign, Inc. *The Domain Name Industry Brief*. Dec. 2017. URL: http://www.verisign.com/assets/domain-name-report-Q32017.pdf (visited on 02/13/2018).

[W3C17] W3C. *HTML 5.1 2nd Edition - W3C Recommendation*. 2017. URL: https://www.w3.org/TR/html5/sec-forms.html#valid-e-mail-address (visited on 12/14/2017).

[Wet16] Jos Wetzels. “Open Sesame: The Password Hashing Competition and Argon2”. In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 104.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 69

Improving Anonymization Clustering

Florian Thaeter[1] Rüdiger Reischuk[2]

Abstract:

Microaggregation is a technique to preserve privacy when confidential information about individuals shall be used by third parties. A basic property to be established is called k-anonymity. It requires that identifying information about individuals should not be unique, instead there has to be a group of size at least k that looks identical. This is achieved by clustering individuals into appropriate groups and then averaging the identifying information. The question arises how to select these groups such that the information loss by averaging is minimal. This problem has been shown to be NP-hard. Thus, several heuristics called `MDAV` ,`V-MDAV` , ... have been proposed for finding at least a suboptimal clustering.

This paper proposes a more sophisticated, but still efficient strategy called `MDAV`* to construct a good clustering. The question whether to extend a group locally by individuals close by or to start a new group with such individuals is investigated in more depth. This way, a noticeable lower information loss can be achieved which is shown by applying `MDAV`* to several established benchmarks of real data and also to specifically designed random data.

Keywords: Microdata anonymization; k-Anonymity; Microaggregation; group clustering

1 Introduction

We consider databases $\mathcal{X}$ containing n individuals that are characterized by quasi-identifiers and confidential attributes. Quasi-identifiers deliver information that can identify a person, for example date and location of birth. Confidential attributes contain sensitive information about a person, for example the amount of his income or his current diseases that should not be disclosed, more precisely should not be linked to an individual.

In order to discuss algorithmic solutions to the anonymization problem, a precise mathematical model for this setting is needed that will be given first. For a sequence of m quasi-identifiers the set of possible values is given by a cartesian product $QI := QI_1 \times \cdots \times QI_m$, where QI_i denotes the values the i-th quasi-identifier can take. Similarly, for a sequence of p confidential attributes we define $CA := CA_1 \times \cdots \times CA_p$. The database $\mathcal{X}$ consists of n records

[1] Universität zu Lübeck, Institut für Theoretische Informatik, Ratzeburger Allee 160, 23562 Lübeck, Deutschland
thaeter@tcs.uni-luebeck.de
[2] Universität zu Lübeck, Institut für Theoretische Informatik, Ratzeburger Allee 160, 23562 Lübeck, Deutschland
reischuk@tcs.uni-luebeck.de

doi:10.18420/sicherheit2018_05

$(\mathbf{x}_i, \mathbf{y}_i) \in QI \times CA$ with $i \in [1 \dots n]$, where n denotes the number of individuals. The vector $\mathbf{x}_i := (x_i^1, x_i^2, \dots, x_i^m) \in QI$ denotes the quasi-identifiers and $\mathbf{y}_i := (y_i^1, y_i^2, \dots, y_i^p) \in CA$ the confidential attributes of the i-th individual. Restricting $\mathcal{X}$ to the quasi-identifiers we write $\mathcal{X}_{QI} := (\mathbf{x}_1, \dots, \mathbf{x}_n)$ and analogously $\mathcal{X}_{CA} := (\mathbf{y}_1, \dots, \mathbf{y}_n)$ for the confidential attributes. The removal of the i-th individual from $\mathcal{X}$ will be denoted by $\mathcal{X} - (\mathbf{x}_i, \mathbf{y}_i)$.

Such non-public databases may be used by third parties to investigate relations between quasi-identifiers and confidential attributes, for example how the age of a person has influence on his diseases. But the database owner should not simply provide all the tuples $(\mathbf{x}_i, \mathbf{y}_i)$ with their real values because this could violate the privacy of the individuals. Instead, the data has to be anonymized first which is done by an anonymization algorithm μ.

Definition 1.1. Let $\mathcal{X}_\mathbf{x} := \{i \mid \mathbf{x}_i = \mathbf{x}\}$ be the index set of all individuals in $\mathcal{X}$ whose QI value is $\mathbf{x}$. A database $\mathcal{X}$ is k-anonymous if for all $\mathbf{x} \in QI \quad |\mathcal{X}_\mathbf{x}| \geq k$ or $\mathcal{X}_\mathbf{x} = \emptyset$ [S02].

This condition means that every vector $\mathbf{x} \in QI$ contained in $\mathcal{X}$ has to occur with multiplicity at least k. In the special case $k = n$ all values for the quasi-identifiers have to be identical and thus provide no additional information if the set of individuals in $\mathcal{X}$ is known. Such a database guaranteeing maximum privacy will be called *QI-uniform*.

Definition 1.2. An anonymization algorithm μ is a mapping $\mu : (QI \times CA)^n \to (QI \times CA)^n$,

$$\mu : \mathcal{X} := (\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n) \mapsto \hat{\mathcal{X}} := (\hat{\mathbf{x}}_1, \mathbf{y}_1), \dots, (\hat{\mathbf{x}}_n, \mathbf{y}_n)\,,$$

that changes the values of the quasi-identifiers in a database $\mathcal{X}$ to generate an anonymous database $\hat{\mathcal{X}}$. μ achieves k-anonymity if $\mu(\mathcal{X})$ is k-anonymous for every $\mathcal{X}$. If $\mathcal{X}$ is already QI-uniform we require that μ does not change $\mathcal{X}$ to exclude trivial mappings.

The computation of the new values $\hat{\mathbf{x}}_i$ is often done with the help of a *centering algorithm* c. In case of real-valued or ordered data c might be the arithmetic mean denoted by $c_{\texttt{MEAN}}$ or the median $c_{\texttt{MEDIAN}}$ taken of every coordinate independently. $c_{\texttt{MEAN}}(\mathcal{X})$ is also called the *centroid* of $\mathcal{X}$.

Definition 1.3. A centering algorithm $c : QI^* \to QI$ calculates a vector $\overline{\mathbf{x}} \in QI$ that is supposed to represent the elements of a sequence $\mathbf{x}_1, \mathbf{x}_2, \dots \in QI^*$. For the case of identical vectors we assume that $c(\mathbf{x}, \mathbf{x}, \dots) = \mathbf{x}$.

To evaluate the quality of an anonymization algorithm a metric $d(\mathbf{x}, \mathbf{x}')$ is used on the set QI. We extend d to a metric for two sequences $\mathcal{X}_{QI}, \mathcal{X}'_{QI}$ of equal length with the help of a function $f : \mathbb{R}_+^n \to \mathbb{R}_+$ by $d(\mathcal{X}_{QI}, \mathcal{X}'_{QI}) = f(d(\mathbf{x}_1, \mathbf{x}'_1), \dots, d(\mathbf{x}_n, \mathbf{x}'_n))$. The aggregation function f could, for example, be any ℓ_p-norm for $1 \leq p \leq \infty$.

Definition 1.4. The anonymization distortion of an algorithm μ applied to a database $\mathcal{X}$ is defined by $D_\mu(\mathcal{X}) := d(\mathcal{X}_{QI}, \mu(\mathcal{X})_{QI})$.

Scaling the values of a database $\mathcal{X}$ by a factor $\alpha > 1$ will increase the anonymization distortion, too, except for trivial metrics. Thus, distortion should be measured relative to the data expansion in $\mathcal{X}$ which will be called diversity.

The diversity $\Delta(\mathcal{X}) \in \mathbb{R}_+$ should fulfill the condition: $\Delta(\mathcal{X}) = 0$ iff $\mathcal{X}$ is QI-uniform. For example, $\Delta(\mathcal{X})$ could be the sum of pairwise distances $\sum_{1 \le i < i' \le n} d(\mathbf{x}_i, \mathbf{x}_{i'})$ denoted by $\Delta_{\mathtt{PD}}$. Alternatively, one could use the sum of distances to a center of the whole sequence given by $\Delta_c := \sum_i d(\mathbf{x}_i, c(\mathcal{X}_{QI}))$. Then the information loss is defined as the quotient of anonymization distortion and diversity.

Definition 1.5. The information loss $L_\mu(\mathcal{X})$ when applying an anonymization algorithm μ to a non-QI-uniform database $\mathcal{X}$ is defined as $L_\mu(\mathcal{X}) := D_\mu(\mathcal{X})/\Delta(\mathcal{X})$.

The anonymization technique considered in this paper is called *microaggregation* [D09]. Given k it creates a *k-clustering* of a sequence $\mathcal{X}$ that is described by a partition $\mathcal{G} := G_1 \cup G_2 \cup \cdots \cup G_t$ of the indices of the elements of $\mathcal{X}$ into groups G_ℓ such that $|G_\ell| \ge k$ for every ℓ. Let us denote the elements of a group G_ℓ by $\ell_1, \ldots, \ell_{|G_\ell|}$. We call a k-clustering *strict* if each group contains exactly k elements except at most one (in case the size of $\mathcal{X}$ is not a multiple of k).

After the partitioning for every group G_ℓ a representative vector $\overline{\mathbf{x}}_\ell = c(\mathbf{x}_{\ell_1}, \ldots, \mathbf{x}_{\ell_{|G_\ell|}})$ is created by applying a centering algorithm c. In the anonymized output $\hat{X}$ each vector $\mathbf{x}$ belonging to group G_ℓ is replaced by its corresponding group representative $\overline{\mathbf{x}}_\ell$. Obviously, microaggregation achieves k-anonymity.

For microaggregation there are specific choices for the functions c, d, f, Δ which are commonly used for real-valued quasi-identifiers [DMS06, DM02a, DSS08, LM05, SMD06]. The *squared euclidean distance* of two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$ is defined as $d_{\mathtt{SSE}}(\mathbf{x}, \mathbf{x}') := \sum_j (x_j - x'_j)^2$. Now if f is chosen as the sum operator, for two sequences $\mathcal{X}, \hat{\mathcal{X}}$ the value $d_{\mathtt{SSE}}(\mathcal{X}, \hat{\mathcal{X}})$ is called the *sum of squared errors*. This defines the anonymization distortion

$$D_\mu^{\mathtt{SSE}}(\mathcal{X}) \;:=\; \sum_{i=1}^{n} d_{\mathtt{SSE}}(\mathbf{x}_i, \mu(\mathcal{X})_i) \;.$$

For a k-clustering with groups $G_1, \ldots, G_t$ this can be rewritten as $\sum_{\ell=1}^{t} d_{\mathtt{SSE}}(G_\ell)$ where $d_{\mathtt{SSE}}(G_\ell) := \sum_{i \in G_\ell} d_{\mathtt{SSE}}(\mathbf{x}_i, \overline{\mathbf{x}}_\ell)$. In this case it can be shown

Lemma 1.6. If the representative $\overline{\mathbf{x}}_\ell$ of a group G_ℓ consisting of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{|G_\ell|}$ is chosen as $c_{\mathtt{MEAN}}(\mathbf{x}_1, \ldots, \mathbf{x}_{|G_\ell|})$ then $d_{\mathtt{SSE}}(G_\ell)$ is minimized.

This motivates to measure the diversity of a database $\mathcal{X}$ by considering it as a single group and to measure the individual SSE-distances to a center $\bar{\mathbf{x}}$ for the whole group that is computed by $\bar{\mathbf{x}} := c_{\mathtt{MEAN}}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. Thus, we define $\Delta^{\mathtt{SSE}}(\mathcal{X}) := \sum_{i=1}^{n} d_{\mathtt{SSE}}(\mathbf{x}_i, c_{\mathtt{MEAN}}(\mathbf{x}_1, \ldots, \mathbf{x}_n))$. For this setting it can be shown:

Lemma 1.7. $\sum_{\ell=1}^{t} d_{\mathrm{SSE}}(G_\ell) \leq \Delta^{\mathrm{SSE}}(\mathcal{X})$ for any partition of $\mathcal{X}$ into groups G_ℓ.

As a consequence, the information loss $L_\mu^{\mathrm{SSE}}(\mathcal{X}) := D_\mu^{\mathrm{SSE}}(\mathcal{X})/\Delta^{\mathrm{SSE}}(\mathcal{X})$ is normalized to the interval $[0, 1]$ for every microaggregation algorithm μ.

The obvious optimization problem for microaggregation is to find a k-clustering into groups G_ℓ that minimizes $\sum_{\ell=1}^{t} d_{\mathrm{SSE}}(G_\ell)$. It has been observed that there always exists an optimal clustering with group sizes between k and $2k - 1$ [DM02a]. Still, finding an optimal clustering remains a computationally difficult problem.

Theorem 1.8. [OD01] Optimal microaggregation for $m \geq 2$ and $k = 3$ is NP-hard for the metric d_{SSE}.

This claim has also be made for $k > 3$ without giving a proof. For $k = 2$ finding an optimal strict clustering can be reduced to the weighted matching problem and thus is efficiently solvable. In the nonstrict case, in an optimal clustering every group can consist of 2 or 3 elements and its computational complexity seems to be open. It is also unclear how well this problem can be approximated. Therefore, good heuristics for arbitrary k are of interest.

The rest of this paper is structured as follows. In section 2 we discuss the standard microaggregation heuristic `MDAV` and two important variants of it. Next in section 3 we present a new strategy for k-clustering called `MDAV`* . This section also contains a concrete example of a simple instance where `MDAV`* outperforms the other algorithms by far. Its complexity is analyzed in section 4. Experimental results on established benchmark databases are presented in section 5 that illustrate the improvements achieved.

2 MDAV

The two most common microaggregation algorithms for k-anonymity are `MDAV` [DM02a][DT05][D09] and the more recent PCL [RFP13]. `MDAV` (maximum distance to average vector) relies on a greedy nearest-neighbor aggregation technique and generates a strict k-clustering, whereas PCL uses a modification of the Lloyd algorithm for aggregation. PCL achieves lower information losses than `MDAV` on synthetical as well as on standard data sets, but this comes at the cost of a substantially longer running time [RFP13]. There are several variations of `MDAV` that differ slightly in time complexity and information loss obtained. We use the `MDAV` specification from [D09].

[SMD06] applies several simplifications and improvements to `MDAV`. Instead of forming two groups at extremal regions simultaneously only a single group is constructed at a time, the global centroid is not recomputed each time, and leftovers at the end are assigned to their closest groups instead of forming a new group out of them. This variant denoted by `MDAV`$^+$ is formally defined below.

Algorithm MDAV$^+$

1. Compute the centroid $\overline{\mathbf{x}}$ of the input dataset $\mathcal{X}$.
2. Select an unassigned record $\mathbf{x}_r$ furthest away from $\overline{\mathbf{x}}$.
3. Form a group around $\mathbf{x}_r$ consisting of $\mathbf{x}_r$ and its $k-1$ closest unassigned neighbors (these elements are now assigned).
4. If there are at least k unassigned records left go back to step 2, otherwise put each not yet assigned element into its closest group.

A significant improvement of MDAV is V-MDAV (Variable group size MDAV) that can generate nonstrict k-clusterings. If a region contains more than k elements, MDAV splits it to obtain groups of fixed size k. V-MDAV solves this problem by considering the inclusion of additional records to newly formed groups.

Let U be the index set of all unassigned records and G be the most recently established group. If the size of G is smaller than $2k-1$ we select a pair of elements $(\mathbf{x}_{\bar{i}}, \mathbf{x}_{\bar{j}})$ with $\bar{i} \in U$ and $\bar{j} \in G$ that minimizes $d_{\mathrm{SSE}}(\mathbf{x}_i, \mathbf{x}_j)$ over all $(i,j) \in U \times G$. Denote this value by $d_{in} := d_{\mathrm{SSE}}(\mathbf{x}_{\bar{i}}, \mathbf{x}_{\bar{j}})$. It is compared to the distance of $\mathbf{x}_{\bar{i}}$ to the closest unassigned element

$$d_{out} := \min_{i \in U - \bar{i}} d_{\mathrm{SSE}}(\mathbf{x}_{\bar{i}}, \mathbf{x}_i).$$

To decide, how to handle $\mathbf{x}_{\bar{i}}$ V-MDAV uses a gain factor γ. $\mathbf{x}_{\bar{i}}$ is added to G iff $d_{in} < \gamma \cdot d_{out}$, otherwise a new group is established around $\mathbf{x}_{\bar{i}}$. For $\gamma = 0$ V-MDAV equals MDAV$^+$. With $0 < \gamma \leq 1$, $\mathbf{x}_{\bar{i}}$ can extend G only if it is closer to G than to its closest unassigned neighbor. For $\gamma > 1$ V-MDAV favors to assign $\mathbf{x}_{\bar{i}}$ to G even if there might be a closer unassigned neighbor. The last decision seems only reasonable for $k > 2$. As stated in [SMD06] it is not clear how an optimal choice of γ should look like. The authors recommend $\gamma = 0.2$ for scattered data sets and $\gamma = 1.1$ for grouped data sets.

3 MDAV*

The main novelty of the heuristic MDAV* is to take into account the effects on nearby records when the extension of a group has to be decided and to handle group extension before creating a new group instead of after the creation. When assigning elements to groups we consider the additional costs per element (marginal costs) a decision would cause and (greedily) select an optimal one.

Before we go into details, consider the simple example of a database $\mathcal{X}$ with a single quasi identifier attribute. It contains 11 records depicted by values $\mathbf{x} \in \{1, 2, 3, 5, 6, 19, 20, 21, 98, 99, 100\}$ of this attribute. The centroid of $\mathcal{X}$ is 34 resulting in diversity $\Delta^{\mathrm{SSE}} = 17966$. For the anonymity parameter we choose $k = 3$.

To construct a k-clustering of $\mathcal{X}$ the heuristic V-MDAV starts with a group G_1 containing the values 98, 99 and 100. Because there are no unassigned elements near that group G_1 does not get expanded. Then the group G_2 with elements 1, 2 and 3 is created – similarly this group will not be expanded because $\mathbf{x}_{\bar{i}} = 5$ is chosen, which results in $d_{in} = d(3,5) = 4$ and $d_{out} = d(5,6) = 1$, thus 5 is not included in G_2. The third group G_3 is created out of 5 and contains 5, 6 and 19. Finally, the elements 20 ($d_{in} = d(19,20) = 1, d_{out} = d(20,21) = 1$) and 21 ($d_{in} = d(20,21), d_{out} = \infty$) join this group. The centroids of these groups are 2, 14.2 and 99 yielding an anonymization distortion of

$$D^{\mathrm{SSE}}_{\text{V-MDAV}} = d_{\mathrm{SSE}}(G_1) + d_{\mathrm{SSE}}(G_2) + d_{\mathrm{SSE}}(G_3) = 2 + 2 + 254.8 = 258.8$$

and information loss $L^{\mathrm{SSE}}_{\text{V-MDAV}} \approx 1.44\%$.

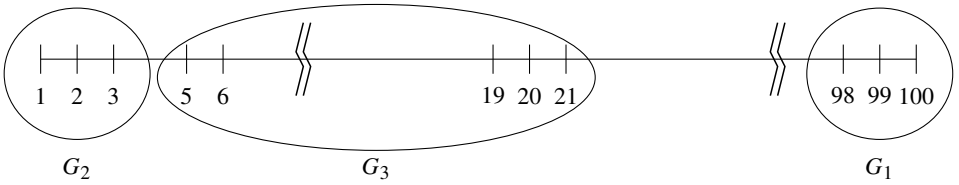


Fig. 1: 3-Clustering generated by V-MDAV

Creation of the group G_3 with 5 elements from 5 to 21 seems to be a bad decision. It is caused by the fact that 5 has a close neighbor 6 to open a new group. However, this should not be considered as reason enough not to include 5 in the already established close group G_2. It would be better to compare the consequences of including 5 in G_2 to building a new group out of 5. This fact brings us back to MDAV*.

The state of MDAV* can be described by a partitioning $\mathcal{G} := G_1 \cup \cdots \cup G_t \cup U$, which consists of t disjoint groups G_ℓ of size at least k and an additional group U containing all records which are still unassigned. Given the state of the algorithm we define the neighborhood $N_q(\mathbf{x}, G) \subseteq G$ as the subset of a group G, containing the indices of $\mathbf{x}$ and its q closest neighbors. Furthermore, $\mathrm{clos}(\mathbf{x}) \in \mathcal{G} \setminus \{U\}$ denotes a group G that is closest to $\mathbf{x}$. If an element $\mathbf{x}$ is included in a group G, we write $G + \mathbf{x}$, if it is removed $G - \mathbf{x}$.

After the choice of a new cluster origin $\mathbf{x}_r \in U$, MDAV* considers two options. The first one is to build a new group $N_{k-1}(\mathbf{x}_r, U)$ as usual. The second one is to extend $\mathrm{clos}(\mathbf{x}_r)$ by $\mathbf{x}_r$ in which case the group $N_{k-1}(\mathbf{x}_r, U)$ cannot be built. Instead the rest of $N_{k-1}(\mathbf{x}_r, U)$ has to be assigned somehow differently. For this, we take the closest neighbor $\mathbf{y} \in U$ of $\mathbf{x}_r$ and consider establishing a new group around $\mathbf{y}$. The underlying decision rule considers the marginal costs in both cases. Still, this is only an estimate of a best possible usage of $\mathbf{x}_r$ because we do not know whether $\mathbf{y}$ should ever be chosen as the origin of a new group.

The costs divided by the number k of elements for creating a new group $N_{k-1}(\mathbf{x}_r, U)$ out of record $\mathbf{x}_r$ are

$$\frac{d_{\mathrm{SSE}}(N_{k-1}(\mathbf{x}_r, U))}{k} \tag{1}$$

while the costs per element of extending the group $\text{clos}(\mathbf{x}_r)$ by $\mathbf{x}_r$ and establishing a new group around $\mathbf{y}$ (now assigning $k+1$ elements) are

$$\frac{d_{\text{SSE}}(\text{clos}(\mathbf{x}_r) + \mathbf{x}_r) - d_{\text{SSE}}(\text{clos}(\mathbf{x}_r)) + d_{\text{SSE}}(N_{k-1}(\mathbf{y}, U - \mathbf{x}_r))}{k+1}. \tag{2}$$

Algorithm MDAV*

1. Compute the centroid $\overline{\mathbf{x}}$ of the input dataset $\mathcal{X}$ and initialize U with $\mathcal{X}$.
2. Select $\mathbf{x}_r \in U$ furthest away from $\overline{\mathbf{x}}$.
3. Compute $N_{k-1}(\mathbf{x}_r, U)$ and the group $\text{clos}(\mathbf{x_r})$.
4. Based on the marginal costs (1) and (2) choose between
   • extending $\text{clos}(\mathbf{x_r})$ by $\mathbf{x}_r$ and removing $\mathbf{x}_r$ from U versus
   • establishing the new group $N_{k-1}(\mathbf{x}_r, U)$ and removing these elements from U.
5. If $|U| \geq k$ go back to step 2,
   otherwise assign each $\mathbf{x} \in U$ to $\text{clos}(\mathbf{x})$.

We are now able to describe how MDAV* handles the situation from above. After creating group G_1 and G_2 as V-MDAV does, considering the next element 5 it is included in G_1, because the costs for creating a new group $N_2(\mathbf{x}_r, U)$ (containing $5, 6$ and 19) are $40.\overline{6}$ whereas the costs for expanding the group G_2 are 32.19. Also 6 is included in G_2, because the costs for a new group are $40.\overline{6}$ and costs for expanding G_2 are only 2.6125. The centroid of G_2 becomes 3.4 and results in a distortion

$$D^{\text{SSE}}_{\text{MDAV}^*} = d_{\text{SSE}}(G_1) + d_{\text{SSE}}(G_2) + d_{\text{SSE}}(G_3) = 2 + 17.2 + 2 = 21.2$$

and information loss $L^{\text{SSE}}_{\text{MDAV}^*} \approx 0.118\%$. Now, comparing $L^{\text{SSE}}_{\text{V-MDAV}}$ and $L^{\text{SSE}}_{\text{MDAV}^*}$ we see that for this instance the information loss of V-MDAV is more than 12 times larger than that of MDAV*.

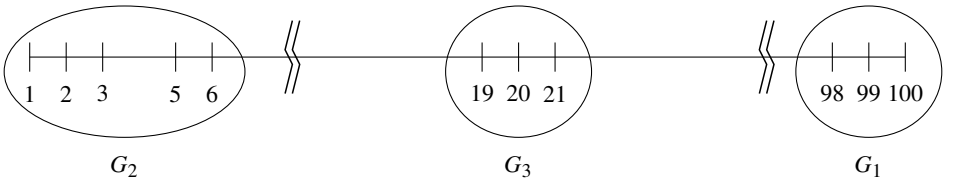


Fig. 2: 3-Clustering generated by MDAV*

An important part missing so far is the calculation of $\text{clos}(\mathbf{x}_r)$. In V-MDAV the distance between a group G and a record $\mathbf{x}$ is measured by the distance between $\mathbf{x}$ and the record from G closest to $\mathbf{x}$. Our goal is to increase d_{SSE} of a group as little as possible when $\mathbf{x}$ is included. However, the increase of group SSE depends on the group's size. This implies that this shortcut may deliver a non-optimal record when searching for the best group to extend with a fixed record.

Fact 3.1. The distance measure between a group and a record used by V-MDAV depends on the group size and is not optimal in general.

To address this problem, MDAV* chooses clos($\mathbf{x}$) by looking directly at the growth of d_{SSE} the extension would cause:

$$\mathrm{clos}(\mathbf{x}) := \operatorname*{arg\,min}_{\mathcal{G}\setminus\{U\}} d_{\mathrm{SSE}}(G + \mathbf{x}) - d_{\mathrm{SSE}}(G) \ .$$

4 Complexity analysis

MDAV$^+$ requires $\lfloor n/k \rfloor$ rounds to create that many groups of size k each except possibly the last one. Computing $\mathbf{x}_r$ and its $(k-1)$-neighborhood can be done in time $O(k\ n)$. This gives a total time complexity of $O(n^2)$.

V-MDAV and MDAV* may run for almost n rounds if the extension of a group by a single element happens often. d_{in} and d_{out} can be computed in $O(k\ n)$ steps which results in an upper time bound $O(k\ n^2)$ for V-MDAV.

To compute clos($\mathbf{x}$) for at most n/k groups their distortion has to be recomputed which requires $O(k)$ steps per group. For the $(k-1)$-neighborhoods time $O(k\ n)$ suffices as in MDAV$^+$. Again this gives the bound $O(k\ n^2)$ for MDAV*.

If we consider k as a constant then the asymptotic growth of the time complexity of these three algorithms is quadratic with respect to the size of the database. This can even be lowered if we order the elements in U with respect to their distance to the centroid and in each round perform a bounded search for closest neighbors.

It remains to compare the performance of these heuristics with respect to the information loss. Since the optimization problem has shown to be hard it will be difficult to compute the minimum possible distortion achievable by clustering. Thus, there is little hope to derive performance bounds analytically in general. To get some insight in how the heuristics perform in practice we have conducted a bunch of experiments.

5 Information loss – experimental results

The information loss has been estimated on different kinds of data. On the one hand, three benchmark data sets (CENSUS, TARRAGONA and EIA) from the CASC project [DM02b] have been used. CENSUS contains 13 numerical attributes and 1080 records. It was created using the Data Extraction System of the U.S. Bureau of Census in 2000. TARRAGONA contains 13 numerical attributes and 834 records. It contains data of the Spanish region Tarragona from 1995. The EIA data set consists of 15 attributes and 4092 records. As in

[DMS06] we only use a subset of 11 attributes precisely 1 and 6 to 15 and ignore categorical data in attribute 2 and 3 as well as attributes with small width in attributes 4 and 5.

We have also tested the information loss on synthetic data. For this process the uniform data set SimU and the grouped data set SimC as in [DMS06] have been created. SimU contains 1000 records with 10 independent numerical attributes which values are chosen uniformly at random. SimC contains clustered data. First 100 cluster centers with 10 attributes are chosen like in SimU. Then for each cluster a number in the interval $[4 \ldots 21]$ is randomly chosen as its size and the cluster is filled with that many elements differing from the cluster center by at most 0.5% in every attribute dimension. Finally $x/3$ independent records are added, where x is the number of records created so far. For uniform as for grouped data we have created 25 data sets and taken the average information loss as final result for all algorithms.

5.1 Test methodology

In all test cases we have standardized the data prior to anonymization. By adjusting the mean value to 0 and the variance to 1 for every attribute the influence by the size of numbers (some attributes have a small range, others a much larger one) has been eliminated. As a consequence no attribute looks more important than others to an anonymization algorithm. L_μ^{SSE} is used on the standardized database $\mathcal{X}$ as information loss measure. The numerical values shown in the tables have been multiplied by 100, thus percentages are listed (the same format as in previous publications). Thus, our results are directly comparable to the results e.g. in [DMS06] or [SMD06]. All tests have been conducted for $k \in \{3, 4, 5, 7, 10\}$. `V-MDAV` has been tested with gain factors γ between 0.0 or 2.0 in steps of 0.1. Only the best result is shown here (see table 4 for the values used in every test case).

5.2 Results

The results are listed in table 1 and table 2 as well as in graphical form in figure 3 to figure 5. As summarized in table 3a) there are only 3 out of 25 test cases in which $\texttt{MDAV}^+$ is slightly better than $\texttt{MDAV}^*$. In all other cases the information loss inflicted by $\texttt{MDAV}^*$ is between 0.9 and 45.4 percent lower than the one of $\texttt{MDAV}^+$ on the same data. On average over the 25 test cases shown the information loss of $\texttt{MDAV}^*$ is about 7.5% lower.

In table 3b) $\texttt{MDAV}^*$ is compared to `V-MDAV`. Because `V-MDAV` can behave like $\texttt{MDAV}^+$ (setting $\gamma = 0$) it can never be worse than $\texttt{MDAV}^+$ if for every instance the (unknown) optimal scaling factor is used. The results show that there are scenarios in which `V-MDAV` takes profit from this. However, in most cases $\texttt{MDAV}^*$ has an even lower information loss than `V-MDAV`. This clearly shows the further improvement achieved by $\texttt{MDAV}^*$.

Another interesting question is how the information loss increases with k. For our data sets this has a significant impact. In figure 6 the impact of k for the SimC test in the range $k \in \{2, 3, \ldots, 100\}$ is shown. The graphs for other test cases look similar and are omitted here.

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
MDAV	5.677	7.537	9.056	11.608	14.186
MDAV$^+$	5.662	7.514	9.007	11.657	14.073
V-MDAV	5.662	7.514	8.978	11.586	14.043
MDAV*	5.782	7.433	8.809	11.369	14.003

(a) CENSUS

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
MDAV	16.922	19.540	22.459	27.525	33.195
MDAV$^+$	16.951	19.767	22.872	28.255	33.254
V-MDAV	15.849	19.695	22.872	28.249	33.251
MDAV*	16.143	19.189	22.250	28.399	34.743

(b) TARRAGONA

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
MDAV	0.483	0.672	1.668	2.187	3.846
MDAV$^+$	0.488	0.673	1.775	2.211	3.547
V-MDAV	0.465	0.673	1.056	2.211	2.794
MDAV*	0.449	0.617	0.911	2.032	2.633

(c) EIA

Tab. 1: Information loss on standard test data sets given in percentages ($100 \cdot L_\mu^{\mathrm{SSE}}$)

6 Future work

Can the clustering problem be solved by an approximation algorithm with a guaranteed approximation rate for the information loss? The only result known to us is an algorithm in [DSS08] with a quite high rate of $O(k^3)$. It should be possible to improve this bound significantly. The simple 1-dimensional example given above illustrates that in the worst case MDAV* might be much better than V-MDAV. It would be interesting to prove a bound on the approximation ratio of MDAV* or a further improved strategy.

The property k-anonymity has been extended to stronger privacy requirements called ℓ-diversity and t-closeness. However, these properties seem to induce even higher information loss and algorithmic solutions are significantly more difficult to analyze in a rigorous way. Can one establish any performance guarantees for these settings?

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
MDAV	18.181	23.656	28.072	34.671	40.826
MDAV$^+$	18.026	23.618	27.897	34.043	40.625
V-MDAV	17.993	23.551	27.803	34.006	40.427
MDAV*	17.756	23.095	27.280	33.371	39.643

(a) SimU

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
MDAV	6.957	9.294	11.246	14.511	18.436
MDAV$^+$	6.856	9.233	11.045	14.114	18.086
V-MDAV	6.192	8.355	10.155	13.142	16.973
MDAV*	5.999	8.163	9.659	12.315	15.683

(b) SimC

Tab. 2: Information loss on synthetic test data sets given in percentages ($100 \cdot L_\mu^{\mathrm{SSE}}$)

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
CENSUS	+1.8%	−1.4%	−2.7%	−2.1%	−1.3%
TARRAGONA	−4.6%	−1.8%	−0.9%	+3.2%	+4.7%
EIA	−7.0%	−8.1%	−45.4%	−7.1%	−31.5%
SimU	−2.3%	−2.4%	−2.8%	−3.7%	−2.9%
SimC	−13.8%	−12.2%	−14.1%	−15.1%	−14.9%

(a) MDAV* versus MDAV$^+$

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
CENSUS	+2.11%	−1.08%	−1.89%	−1.87%	−0.29%
TARRAGONA	+1.85%	−2.57%	−2.72%	+0.53%	+4.49%
EIA	−3.40%	−8.23%	−13.71%	−8.09%	−5.77%
SimU	−1.32%	−1.93%	−1.88%	−1.86%	−1.94%
SimC	−3.11%	−2.29%	−4.88%	−6.30%	−7.60%

(b) MDAV* versus V-MDAV

Tab. 3: Percental information loss difference of MDAV* compared to MDAV (a) and V-MDAV (b) Note that negative numbers show an improvement.

	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$
CENSUS	$\gamma = 0.0$	$\gamma = 0.0$	$\gamma = 0.2$	$\gamma = 0.1$	$\gamma = 0.2$
TARRAGONA	$\gamma = 0.3$	$\gamma = 0.3$	$\gamma = 0.0$	$\gamma = 0.6$	$\gamma = 0.3$
EIA	$\gamma = 0.6$	$\gamma = 0.0$	$\gamma = 0.4$	$\gamma = 0.0$	$\gamma = 1.3$
SimU	$\gamma = 0.176$	$\gamma = 0.208$	$\gamma = 0.232$	$\gamma = 0.108$	$\gamma = 0.244$
SimC	$\gamma = 0.368$	$\gamma = 0.456$	$\gamma = 0.528$	$\gamma = 0.660$	$\gamma = 0.776$

Tab. 4: Optimal gain factors used for V-MDAV in the experiments. For SimU and SimC the arithmetic mean of the optimal γ for each of the 25 sets is shown.

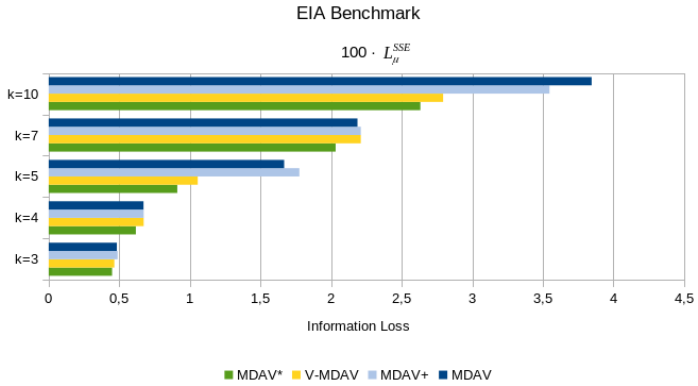


Fig. 3: Information loss for EIA benchmark

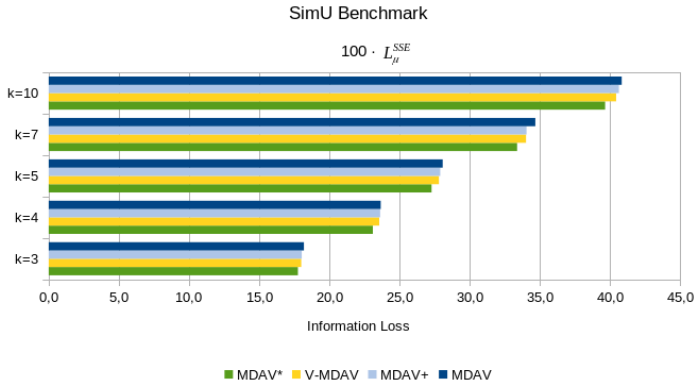


Fig. 4: Information loss for SimU benchmark

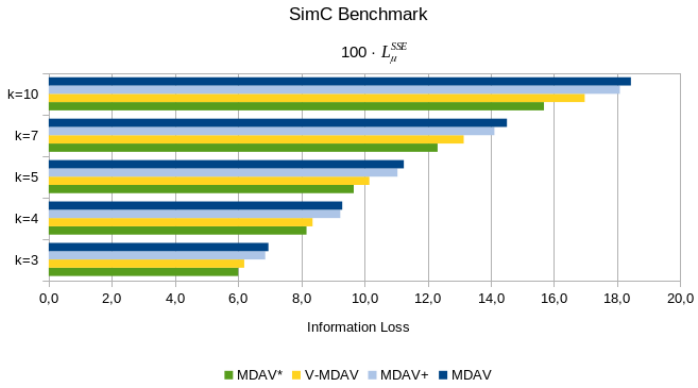


Fig. 5: Information loss for SimC benchmark

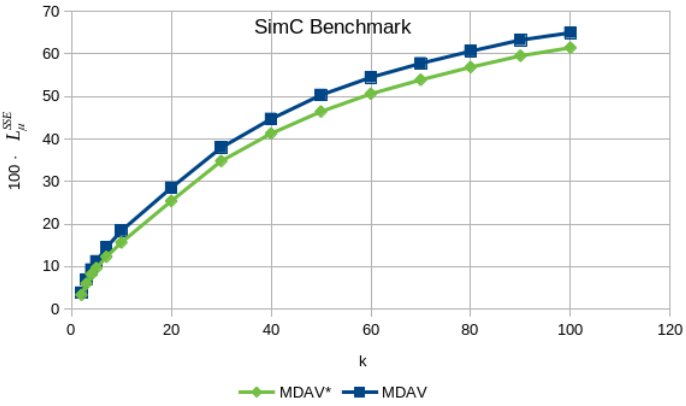


Fig. 6: Information loss tendency for different k

References

[D09] Domingo-Ferrer, J.: Encyclopedia of Database Systems. Springer US, chapter Microaggregation, pp. 1736–1737, 2009.

[DM02a] Domingo-Ferrer, J.; Mateo-Sanz, J. M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and data Engineering, 14(1):189–201, 2002.

[DM02b] Domingo-Ferrer, J.; Mateo-Sanz, J. M.: , Reference data sets to test and compare sdc methods for protection of numerical microdata. `http://neon.vb.cbs.nl/casc`, 2002.

[DSS08] Domingo-Ferrer, J.; Sebé, F.; Solanas, A.: A polynomial-time approximation to optimal multivariate microaggregation. Computers & Mathematics with Applications, 55(4):714–732, 2008.

[DT05] Domingo-Ferrer, J.; Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Mining and Knowledge Discovery, 11(2):195–212, 2005.

[DMS06] Domingo-Ferrer, J.; Martínez-Ballesté, A.; Mateo-Sanz, J. M.; Sebé, F.: Efficient multivariate data-oriented microaggregation. The VLDB Journal—The International Journal on Very Large Data Bases, 15(4):355–369, 2006.

[LM05] Laszlo, M.; Mukherjee, S.: Minimum spanning tree partitioning algorithm for microaggregation. IEEE Transactions on Knowledge and Data Engineering, 17(7):902–911, 2005.

[OD01] Oganian, A.; Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Statistical Journal of the United Nations Economic Commission for Europe, 18(4):345–353, 2001.

[RFP13] Rebollo-Monedero, D.; Forné, J.; Pallarès, E.; Parra-Arnau, J.: A modification of the Lloyd algorithm for k-anonymous quantization. Information Sciences, 222:185–202, 2013.

[SMD06] Solanas, A.; Martinez-Balleste, A.; Domingo-Ferrer, J.: V-MDAV: a multivariate microaggregation with variable group size. In: 17th COMPSTAT Symposium of the IASC, Rome. pp. 917–925, 2006.

[S02] Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557–570, 2002.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 83

Verbesserung der Syndrome-Trellis-Kodierung zur Erhöhung der Unvorhersagbarkeit von Einbettpositionen in steganographischen Systemen

Olaf Markus Köhler[1], Cecilia Pasquini[2,4] und Rainer Böhme[3,4]

Abstract: Beim Einbetten einer versteckten Nachricht in ein Trägermedium wählen adaptive steganographische Systeme die Einbettpositionen abhängig von der erwarteten Auffälligkeit der Änderungen. Die optimale Auswahl kann statistisch modelliert werden. Wir präsentieren Ergebnisse einer Reihe von Experimenten, in denen untersucht wird, inwiefern die Auswahl durch Syndrome-Trellis-Kodierung dem Modell unabhängiger Bernoulli-verteilter Zufallsvariablen entspricht. Wir beobachten im Allgemeinen kleine Näherungsfehler sowie Ausreißer an Randpositionen. Bivariate Abhängigkeiten zwischen Einbettpositionen ermöglichen zudem Rückschlüsse auf den verwendeten Kode und seine Parameter. In Anwendungen, welche die Ausreißer nicht mithilfe zufälliger Permutationen verstecken können, kann die hier vorgeschlagene „outlier corrected"-Variante verwendet werden um die steganographische Sicherheit zu verbessern. Die aggregierten bivariaten Statistiken sind dahingegen invariant unter Permutationen und stellen, unter der Annahme mächtiger Angreifer, ein bisher nicht erforschtes Sicherheitsrisiko dar.

Keywords: Steganographie, Syndrome-Trellis-Kodierung

1 Einleitung

Um steganographisch versteckte Nachrichten möglichst unentdeckbar zu halten, ist es von Nutzen, die Positionen von Änderungen am Trägermedium entsprechend ihrer erwarteten Auffälligkeit zu wählen. Damit der Empfänger zum Extrahieren der Nachricht die Positionen der Änderungen nicht rekonstruieren muss, oder ihm diese auf anderem Wege mitgeteilt werden müssen, setzen moderne steganographische Systeme auf Syndrom-Kodierung. Die populärste Form ist die Syndrom-Trellis-Kodierung (STC) [FJF11]. Sie erlaubt die Trennung der Entscheidungen *wo* und *wie* das Trägermedium geändert wird und zeichnet sich dabei durch ihre rechnerische Effizienz aus.

[1] Institut für Informatik, Universität Innsbruck, Österreich, olaf.koehler@uibk.ac.at
[2] Institut für Informatik, Universität Innsbruck, Österreich, cecilia.pasquini@uibk.ac.at
[3] Institut für Informatik, Universität Innsbruck, Österreich, rainer.boehme@uibk.ac.at
[4] ebenfalls Institut für Wirtschaftsinformatik, Universität Münster, Deutschland

[5] Die hier präsentierten Forschungsergebnisse wurden umfangreicher und in englischer Sprache im Rahmen der Konferenz IWDW 2017 in Magdeburg veröffentlicht. [KPB17]

doi:10.18420/sicherheit2018_06

Kurzgefasst erhält STC als Eingaben das Trägermedium, die zu versteckende Nachricht und einen Vektor von Änderungswahrscheinlichkeiten je Position des Trägermediums. Daraus erzeugt sie einen binären Vektor, der angibt, an welchen Positionen das Trägermedium geändert werden muss, um die Nachricht einzubetten. Üblicherweise wird diese Ausgabe als Realisation eines Vektors unabhängiger Bernoulli-Zufallsvariablen abstrahiert [BFP11, FJF11, SCF16]. Demgegenüber stehen die Struktur des Kodes und einhergehende Beschränkung der möglichen Lösungen, welche der vorherigen Abstraktion widersprechen. Unser Ziel ist es, diese Diskrepanz statistisch zu untersuchen, mit der Leitfrage: Wie dicht werden die vorgeschriebenen Änderungswahrscheinlichkeiten durch die STC angenähert?

Im Rahmen unserer empirischen Herangehensweise erzeugen wir 150 Millionen Steganogramme[6]. Die statistische Auswertung der experimentell gesammelten Daten gliedert sich in drei Schritte: Aggregierte Momente (Kapitel 2), univariate Statistiken (Kapitel 3) und bivariate Abhängigkeiten (Kapitel 5). Zu den neuen Erkenntnissen gehört unter anderem, dass die reguläre STC, wie sie in akademischen Veröffentlichungen und der Referenzimplementierung [FFJ] beschrieben ist, die geforderten Eigenschaften am Beginn des Kodes nicht erfüllt. Um die ermittelten Ausreißer zu vermeiden, schlagen wir eine modifizierte Kode-Konstruktion namens OC-STC vor (Kapitel 4).

1.1 Systemmodell

Steganographie durch Modifikation eines Trägermediums erzeugt aus einem Trägermedium $\boldsymbol{x} = (x_i)_{i=1,\ldots,n}$ der Länge n ein Steganogramm $\boldsymbol{y} = (y_i)_{i=1,\ldots,n}$. Das Steganogramm enthält die gewünschte Nachricht $\boldsymbol{m} = (m_j)_{j=1,\ldots,\alpha n}$, wobei α das Verhältnis zwischen den Längen von Nachricht und Trägermedium beschreibt. Die Änderungen am Trägermedium werden als binärer Vektor $\boldsymbol{c} = (c_i)_{i=1,\ldots,n} \in \{0,1\}^n$ formalisiert, wobei einzelne Elemente des Änderungsvektors $\boldsymbol{c}$ als Änderungspositionen c_i bezeichnet werden. Ein passendes Steganogramm zu finden, wird als Einbettprozess bezeichnet.

Daneben muss der Einbettprozess, wie er in Abb. 1 dargestellt ist, das Schutzziel der Unerkennbarkeit erfüllen: Die statistische Unterscheidbarkeit zwischen Trägermedien und Steganogrammen ist zu minimieren. Aufbauend auf dem Modell additiver Störung, wird ausgehend vom Trägermedium ein Kostenvektor $\boldsymbol{\varrho} = (\varrho_i)_{i=1,\ldots,n}$ bestimmt. Mithilfe einer Heuristik wird jeder Position des Trägermediums ein positiver Wert ϱ_i zugewiesen, der den Einfluss einer dortigen Änderung auf die statistische Unterscheidbarkeit schätzt. Im Fall regulärer STC [FJF10] bedeutet die Anwendung des Modells additiver Störung: Der Anteil von Änderungsposition i an der gesamten Störung ist durch $c_i \varrho_i$ gegeben, sodass die gesamte Störung d als Summe $d = \sum_{i=1}^{n} c_i \varrho_i$ bestimmt wird.

[6] Als Trägermedien werden 1000 64×64 Bildausschnitte aus dem BOSSBase-Datensatz v1.01 [BFP11] verwendet. Die vollständigen Details des Experimentaufbaus sind [KPB17] zu entnehmen.

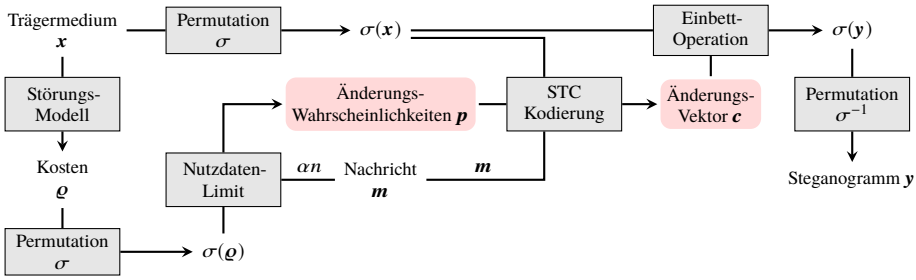


Abb. 1: Systemmodell des Einbettprozesses.

Die Permutation σ wird auf das Trägermedium und den Kostenvektor angewendet. Intuitiv ist die Permutation als Versatz zu begreifen, der die Nachricht in etwa gleichmäßig auf die Positionen des Trägermediums verteilt. Das erhöht die Chance einer erfolgreichen Einbettung, da Positionen mit hohen Kosten oft lokal konzentriert sind. Des Weiteren erhöht die Konvention Passwort-abhängiger pseudozufälliger Permutationen die Sicherheit, da es die Zuordnung von Positionen des Trägermediums zu Nachrichten-Bits erschwert.

Beim Kodierungs-Schritt werden aus dem permutierten Trägermedium $\sigma(\boldsymbol{x})$, der Nachricht $\boldsymbol{m}$ und dem permutierten Kostenvektor $\sigma(\boldsymbol{\varrho})$ die nötigen Änderungen $\boldsymbol{c}$ am Trägermedium bestimmt. Mithilfe der Einbettoperation wird der Änderungsvektor $\boldsymbol{c}$ auf das permutierte Trägermedium $\sigma(\boldsymbol{x})$ angewendet, wodurch das permutierte Steganogramm $\sigma(\boldsymbol{y})$ entsteht. Durch Rückpermutation wird das Steganogramm $\boldsymbol{y}$ erzeugt, was an den Empfänger geschickt wird. Dieser kann die Nachricht durch Permutation des Steganogramms mit σ und Multiplikation mit der Matrix der Syndrom-Kodierung extrahieren. (Das Extrahieren der Nachricht ist in Abb. 1 nicht dargestellt.)

Als Vereinfachung nehmen wir folgend an, dass α so gewählt sei, dass α^{-1} eine Ganzzahl ist. Desweiteren sei $\boldsymbol{x}$ beliebig, aber fest. Zur Vereinfachung der Notationen führen wir die Mehrdeutigkeit ein, dass $\boldsymbol{x}$ und $\boldsymbol{y}$ Ganzzahl-Vektoren im Kontext von Trägermedium und Steganogramm bezeichnen, aber binäre Vektoren im Kontext der Kodierung bezeichnen. Die implizite Annahme dabei ist, dass eine Abbildung von den Ganzzahl-Vektoren auf ihre binäre steganographische Semantik durch die Einbettoperation gegeben ist. In einfachen Fällen ist die Semantik durch das niedrigstwertige Bit gegeben, aber auch andere (sicherere) Einbettoperationen sind möglich.

1.2 Syndrom-Trellis-Kodierung

Bei der Syndrom-Kodierung ergibt sich das Steganogramm $\boldsymbol{y}$ als Syndrom aus dem linearen Gleichungssystem mit der Matrix $\mathbb{H}$ und der Nachricht $\boldsymbol{m}$, $\mathbb{H}\boldsymbol{y} = \boldsymbol{m}$. Die STC [FJF10] ist als Spezialfall der Syndrom-Kodierung zu verstehen, bei dem eine dünn besetzte Matrix $\mathbb{H}$, wie in (1), durch mehrfaches Konkatenieren und nach unten Rücken einer Submatrix

$\hat{\mathbb{H}} \in \{0,1\}^{h \times \alpha^{-1}}$ konstruiert wird. Die Höhe der Submatrix wird als Kodierungsparameter h bezeichnet. Diese Konstruktion erlaubt es mithilfe des Viterbi-Algorithmus [RC12] effizient einen Änderungsvektor $\boldsymbol{c}$ zu finden, der zugleich die gesamte Störung d minimiert und die Nachricht $\boldsymbol{m}$ als Syndrom bedingt, $\mathbb{H}(\boldsymbol{x} \oplus \boldsymbol{c}) = \boldsymbol{m}$.

$$\mathbb{H} = \begin{pmatrix} \hat{\mathbb{H}} & \begin{matrix}0\ldots0\\0\ldots0\end{matrix} & \begin{matrix}0\ldots0\\0\ldots0\end{matrix} & & & \\ \begin{matrix}0\ldots0\\0\ldots0\end{matrix} & \hat{\mathbb{H}} & \hat{\mathbb{H}} & \ddots & 0 & \\ & 0 & & \hat{\mathbb{H}} & \begin{matrix}0\ldots0\\\hat{\mathbb{H}}\end{matrix} & \begin{matrix}0\ldots0\\\vdots\\0\ldots0\end{matrix} \end{pmatrix} . \tag{1}$$

Aufgrund der strikten Minimieren der gesamten Störung d, hängt der Änderungsvektor $\boldsymbol{c}$ von den Kosten $\boldsymbol{\varrho}$ ab. So lassen sich unter dem Modell additiver Störung die Änderungswahrscheinlichkeiten für den Fall einer optimalen Kodierung bestimmen. Die optimale Änderungswahrscheinlichkeit p_i für jede Position i mit den Kosten ϱ_i ist $p_i = e^{-\lambda\varrho_i}\left(1 + e^{-\lambda\varrho_i}\right)^{-1}$ [FF10a, FF10b, FJF11]. Unter der Annahme, dass die Nachricht die volle Entropie erreicht, ist dabei λ so zu wählen, dass die gesamte Entropie der Länge der Nachricht entspricht, $\sum_{i=1\ldots n} H(p_i) = n\alpha$, wobei die binäre Entropie durch $H(p_i) = -p_i \log_2(p_i) - (1-p_i)\log_2(1-p_i)$ gegeben ist.

Der Zufallsvektor der Änderungen sei als $\boldsymbol{C} = (C_i)_{i=1\ldots n}$ mit dem Ergebnisraum $\{0,1\}^n$ gegeben. Aufgrund paarweiser Unabhängigkeit der Änderungspositionen c_i lässt sich die multivariaten Bernoulli-Verteilung von $\boldsymbol{C}$ zu einem Produkt univariater Bernoulli-Verteilungen vereinfachen. Dabei ergibt sich die Wahrscheinlichkeitsfunktion

$$P_{\boldsymbol{C}}(\boldsymbol{c}) = \prod_{i=1,\ldots,n} (p_i)^{c_i}\,(1-p_i)^{1-c_i} \quad . \tag{2}$$

Bei der Einbettung N verschiedener Nachrichten in ein Trägermedium $\boldsymbol{x}$ entstehen N Realisierungen $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)})$ des Zufallsvektors $\boldsymbol{C}$. Bei optimaler Kodierung folgt der Änderungsvektor $\boldsymbol{c}$ der Verteilung (2). Zur Untersuchung der Abweichung von STC zum optimalen Fall definieren wir die relative Häufigkeitsverteilung $\hat{P}_{\boldsymbol{C}} : \{0,1\}^n \to [0,1]$, wobei $\hat{P}_{\boldsymbol{C}}(\boldsymbol{c})$ die relative Häufigkeit der Erscheinung von $\boldsymbol{c}$ unter den beobachteten N Realisierungen angibt.

2 Analyse der Änderungshäufigkeit

Die Anzahl der Änderungen wird als Zufallsvariable $A = \sum_{i=1}^{n} C_i$ formalisiert, wobei die Realisationen von A mit a bezeichnet werden. Unter der Annahme optimaler Kodierung sollte A als Summe unabhängiger Bernoulli-Variablen einer verallgemeinerten Binomialverteilung folgen [Wa93]. Die Verteilung P_A ergibt sich also aus der Summe der Verteilungen der unabhängigen univariaten Bernoulli-Variablen mit Parameter p_i.

Je Trägermedium $\boldsymbol{x}$ wird ein asymmetrisches 95%-Konfidenzintervall $[a_{\min}, a_{\max}]$ bestimmt, sodass $\sum_{a<a_{\min}} P_A(a) \approx 0.025$ und $\sum_{a<a_{\max}} P_A(a) \approx 0.975$. Des Weiteren werden je Trägermedium $\boldsymbol{x}$ und Kodierungsparameter $h \in \{7, 10, 13\}$ unter einer fixen Permutation σ, $N = 50\,000$ zufällige Nachrichten eingebettet. Die sich dabei ergebenden Änderungsvektoren seien als $Z_h = \{\boldsymbol{c}^{(j)}\}_{j=1,\ldots,N}$ gegeben. Zu den Kodierungsparametern passend, werden dabei die Submatrizen der Referenzimplementierung [FFJ] verwendet:

$$\hat{\mathbb{H}}_7 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}^T, \quad \hat{\mathbb{H}}_{10} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^T,$$

$$\hat{\mathbb{H}}_{13} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}^T.$$

Daraufhin bestimmen wir je Trägermedium den Anteil der Änderungshäufigkeiten, die innerhalb des Konfidenzintervalls liegen. Die Anteile sind im Histogramm Abb. 2 aggregiert.

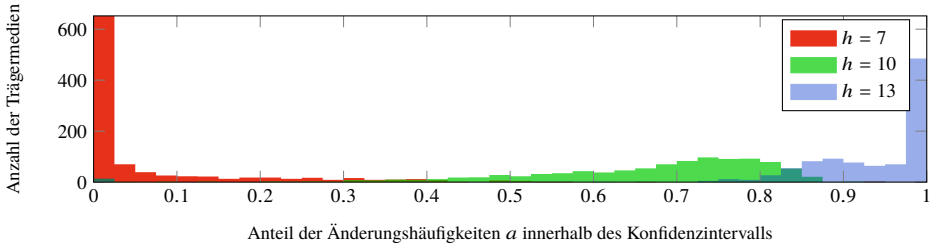


Abb. 2: Histogramm der Änderungsanzahl je Trägermedium innerhalb des 95%-Konfidenzintervalls.

Diskussion der Beobachtungen. Mit dem Kodierungsparameter $h = 13$ ist der mittlere Anteil der Einbettungen im Konfidenzintervall bei 92%. Für kleinere Kodierungsparameter sinkt der Anteil der Einbettungen, die in das Konfidenzintervall fallen. Der für diese Statistik im Falle optimaler Kodierung erwartete mittlere Anteil wäre 95%.

Dass die Zielverteilung nicht vollständig erreicht wird bedeutet, dass zumindest eine der zugrunde liegenden Annahmen nicht zutrifft. Zu diesen Annahmen zählt das Erzielen von Häufigkeiten entsprechend der optimalen Änderungswahrscheinlichkeiten p_i und die Unabhängigkeit der Änderungspositionen c_i. Folgend werden Statistiken untersucht, die es ermöglichen, die Erfüllung dieser beiden Annahmen differenziert zu betrachten.

3 Analyse der Änderungswahrscheinlichkeiten

Zur Untersuchung der individuellen Änderungswahrscheinlichkeiten werden die univariaten Zufallsvariablen C_i betrachtet, welche durch die i-te Komponente des Zufallvektors $\boldsymbol{C}$ gegeben sind. Entsprechend (2) müsste C_i unter der Annahme optimaler Kodierung einer univariaten Bernoulli-Verteilung mit der Wahrscheinlichkeit p_i und der Wahrscheinlichkeitsfunktion $P_{C_i}(c_i) = (p_i)^{c_i}(1-p_i)^{1-c_i}$ folgen.

Für ein beliebiges aber festes Trägermedium $\boldsymbol{x}$ ist die beobachtete Häufigkeit für eine Position i als $\hat{p}_i = \frac{1}{N}\,|\{c \in Z_h : c_i = 1\}|$ gegeben. Zum Vergleich der beobachteten relativen Häufigkeitsverteilung und der optimalen Wahrscheinlichkeitsverteilung von Änderungen genügt es, die Erfolgswahrscheinlichkeiten $\hat{p}_i$ und p_i zu vergleichen, da sie die Bernoulli-Verteilungen vollständig bestimmen.

Ein Vergleich der Änderungswahrscheinlichkeit p_i mit realisierten relativen Häufigkeit $\hat{p}_i$ für ein exemplarisch ausgewähltes Trägermedium, ist in Abbildung 3 (links) dargestellt. Zur Quantifizierung der Abweichung nutzen wir den Hellinger-Abstand $D_{\text{Hellinger}}(p_i,\ \hat{p}_i) = \sqrt{\sqrt{1-p_i}\sqrt{1-\hat{p}_i} + \sqrt{p_i}\sqrt{\hat{p}_i} + 1}$. Die 20 Positionen des Trägermediums mit der stärksten Abweichung sind mit ihrer Position im 4096-elementigen Änderungsvektor $\boldsymbol{c}$ annotiert.

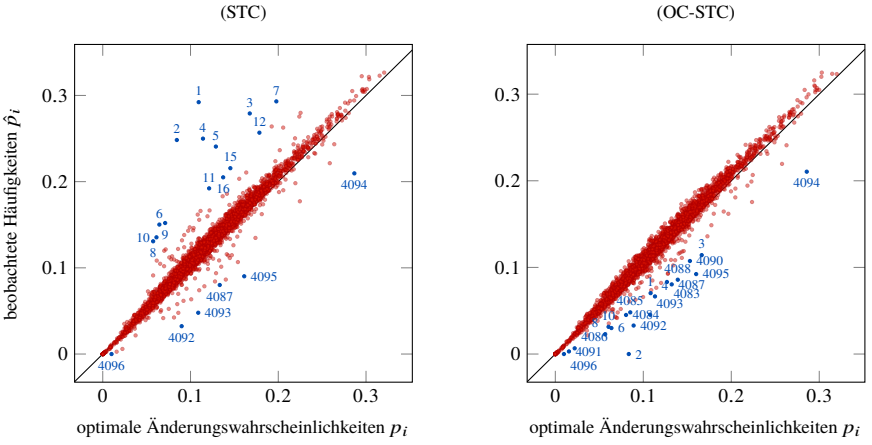


Abb. 3: Exemplarischer Vergleich der optimalen Änderungswahrscheinlichkeiten p_i und beobachteten Häufigkeiten $\hat{p}_i$ mit regulärer STC (links) und der hier vorgeschlagenen OC-STC (rechts). Das zugrundeliegende Trägermedium ist das Graustufenbild 5729 des BOSSBase Datensatzes, ausgeschnitten auf 64×64 Pixel an Position $(258, 53)$. Der verwendete Kodierungsparameter ist $h = 13$.

Diskussion der Beobachtungen. Deutliche Abweichungen von den optimalen Änderungswahrscheinlichkeiten sind besonders am Anfang und Ende des Änderungsvektors $\boldsymbol{c}$ zu beobachten. Dabei fällt auf, dass die Ausreißer am Anfang des Änderungsvektors $\boldsymbol{c}$

im Diagramm über der Diagonale liegen und die Ausreißer am Ende unter der Diagonale. Beide Arten von Ausreißern lassen sich durch die spezifische Konstruktion der Matrix der STC erklären.

Die Ausreißer über der Diagonale lassen sich durch das geringe Hamming-Gewicht in den ersten Zeilen der Matrix $\mathbb{H}$ erklären. Im Fall von $\alpha = 0.5$ hängt das erste Bit der Nachricht, unabhängig vom Kodierungsparameter h, nur von den ersten zwei Bits des permutierten Steganogramms $\sigma(\boldsymbol{y})$ ab. In anderen Worten müssen die ersten zwei Bits des Steganogramms so angepasst werden, dass sie gemeinsam das korrekte erste Bit der Nachricht bilden. Unter der Annahme, dass die Bits des Steganogramms und der Nachricht gleichverteilt sind, ist die Wahrscheinlichkeit, eine Änderung an einer der ersten beiden Positionen vornehmen zu müssen, 50%, obwohl die Summe der optimalen Änderungswahrscheinlichkeiten $p_1 + p_2$ typischerweise deutlich darunter liegt. Ein ähnlicher Effekt trifft die Änderungspositionen $c_j : j < h\alpha^{-1}$, wenn auch weniger ausgeprägt als die ersten beiden Positionen.

Eine der Berechnung optimaler Änderungswahrscheinlichkeiten zugrundeliegende Annahme ist die Einbettung mit maximaler Entropie. Die STC kann dies nur in dem Maß realisieren, in dem die zugrundeliegende Matrix $\mathbb{H}$ dies zulässt. Das deckt sich mit der Beobachtung der im Mittel positiven Abweichung der Häufigkeiten $\hat{p}_i$ von den optimalen Wahrscheinlichkeiten p_i. Kleinere Kodierungsparameter h beschränken die Kodierung stärker und implizieren entsprechend höhere positive Abweichungen der Häufigkeiten.

Diese Beobachtungen lassen sich in der Darstellung Abbildung 3 (links) nachvollziehen. Vergleichbare Ergebnisse sind auch bei den anderen untersuchten Trägermedien zu beobachten. Für eine über alle untersuchten Trägermedien aggregierte Darstellung sei auf [KPB17] verwiesen.

4 OC-STC

Die Ausreißer über der Diagonale entsprechen ungewollt häufiger Änderungen an potentiell auffälligen Positionen und sind damit eine sicherheitsrelevante Abweichung von den optimalen Änderungswahrscheinlichkeiten.

Es ist nicht möglich, die Ausreißer durch eine Anpassung des Kostenvektors $\boldsymbol{\varrho}$ (z.B. mithilfe von einer Fensterfunktion) zu beheben, da die Ausreißer durch die Konstruktion des Kodes bedingt sind. Ein möglicher Ansatz wäre die Detektion derartiger Ausreißer nach der Einbettung und die Korrektur der effektiven Änderungswahrscheinlichkeiten durch die Wiederholung des gesamten Prozesses. Ein solches Vorgehen würde allerdings die Abschätzung des zeitlichen Aufwands der Einbettung erschweren und ähnelt dem unsicheren Verfahren der Steganographie durch Auswahl von Trägermedien.

Stattdessen schlagen wir eine Modifikation der Kode-Konstruktion vor, welche die sicherheitsrelevanten Ausreißer vermeidet. Der Verbesserungsvorschlag wird folgend als OC-STC bezeichnet, was für „Outlier-Corrected Syndrome Trellis Coding“ steht. Die

Kode-Konstruktion von OC-STC ergibt sich durch das Aussparen der ersten $h-1$ Zeilen von $\mathbb{H}$:

$$\mathbb{H}_{\mathrm{OC}} = \begin{pmatrix} \begin{matrix} \\ 0\ldots0 \\ \vdots \\ 0\ldots0 \\ 0\ldots0 \end{matrix} & \ddots & \begin{matrix} \hat{\mathbb{H}} \\ 0\ldots0 \\ 0\ldots0 \end{matrix} & \begin{matrix} \hat{\mathbb{H}} \\ 0\ldots0 \end{matrix} & \begin{matrix} 0\ldots0 \\ \hat{\mathbb{H}} \end{matrix} & & \\ & & & & \ddots & 0 & \\ 0 & & & & \begin{matrix} \hat{\mathbb{H}} \\ \hat{\mathbb{H}} \end{matrix} & \begin{matrix} 0\ldots0 \\ \ddots \end{matrix} & \begin{matrix} 0\ldots0 \\ \vdots \\ 0\ldots0 \end{matrix} \end{pmatrix} . \tag{3}$$

Die dabei resultierende Matrix $\mathbb{H}_{\mathrm{OC}}$ unterscheidet sich in der zentralen Eigenschaft, dass jede Zeile jedes Element von $\hat{\mathbb{H}}$ exakt einmal enthält. Damit ist das Hamming-Gewicht der Zeilen von $\mathbb{H}_{\mathrm{OC}}$ konstant. Somit wird jedes Bit der Nachricht durch gleich viele Positionen des Steganogramms definiert. Auch mit der modifizierten Matrix $\mathbb{H}_{\mathrm{OC}}$ wird die Kodierung mit dem Viterbi-Algorithmus durchgeführt.

Durch das Aussparen der ersten $h-1$ Zeilen wird das Nutzdaten-Limit um $h-1$ Bits reduziert. Um den Einfluss von OC-STC korrekt beurteilen zu können, wird daher ein neuer Wert λ' bestimmt, der die optimalen Änderungswahrscheinlichkeiten entsprechend des reduzierten Nutzdaten-Limits korrekt skaliert. Ein Vergleich der so bestimmten optimalen Änderungswahrscheinlichkeiten und mit OC-STC beobachteten Häufigkeiten ist in Abb. 3 (rechts) dargestellt.

In der exemplarischen Gegenüberstellung zur regulären STC ist die erfolgreiche Vermeidung der Ausreißer über der Diagonale zu erkennen. Die verbleibenden positiven Abweichungen sind dann durch die Beschränkung der Kodierung durch den Kodierungsparameter h zu erklären. OC-STC erzeugt weiterhin Ausreißer unter der Diagonale, was Positionen entspricht, die seltener geändert werden als unter dem verwendeten Modell additiver Störung möglich. Aus diesen Ausreißern ergeben sich nicht unmittelbar Sicherheitsbedenken. Dennoch wäre es wünschenswert, die steganographische Kapazität aller Positionen im Trägermedium voll auszunutzen.

5 Analyse der Abhängigkeiten zwischen Änderungen

Zur Untersuchung paarweiser Abhängigkeiten definieren wir die bivariaten Zufallsvariablen $\boldsymbol{C}_{i,j} = (C_i, C_j)$, die sich aus Komponenten-Paaren von $\boldsymbol{C}$ zusammensetzen. Realisierungen von $\boldsymbol{C}_{i,j}$ werden mit $\boldsymbol{c}_{i,j} = (c_i, c_j)$ bezeichnet. Unter der Annahme optimaler Kodierung wären die Komponenten unabhängig.

Basierend auf Verteilung optimaler Änderungsvektoren (2), ergibt sich für die paarweisen Änderungen $\boldsymbol{C}_{i,j}$ die bivariate Bernoulli-Verteilung mit der Wahrscheinlichkeitsfunktion

$$P_{\boldsymbol{C}_{i,j}}(\boldsymbol{c}_{i,j}) = (p_i)^{c_i}\,(1-p_i)^{1-c_i}\,\left(p_j\right)^{c_j}\,\left(1-p_j\right)^{1-c_j} \quad . \tag{4}$$

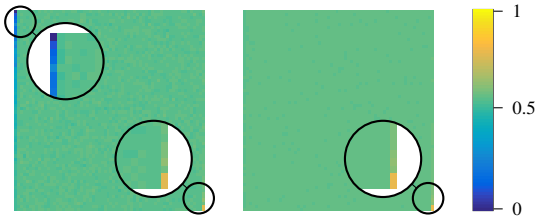


Abb. 4: Spaltenweise Darstellung der gemittelten p-Werte des χ^2-Unabhängigkeitstests über das erste Element im Vergleich zu allen anderen Pixeln j (links), und über jedes Elements j im Vergleich zu allen anderen Elementen (rechts). Die dargestellten Werte sind die über alle Trägermedien gemittelten p-Werte je Position in der Reihenfolge der Kodierung. Der Kodierungsparameter ist $h = 13$.

Wir bestimmen die relativen Häufigkeiten $\hat{p}_{i,j}^{(b'b'')}$, die durch die Rate der beobachteten Änderungsvektoren bestimmt wird. An den Positionen i und j haben diese die binären Werte b' und b''. Daraus lässt sich die empirische Häufigkeitsverteilung $\hat{P}_{\boldsymbol{C}_{i,j}}$ bestimmen. Als bivariate Bernoulli-Verteilung formuliert, ist die Wahrscheinlichkeitsfunktion gegeben als

$$\hat{P}_{\boldsymbol{C}_{i,j}}(\boldsymbol{c}_{i,j}) = \left(\hat{p}_{i,j}^{(00)}\right)^{(1-c_i)(1-c_j)} \left(\hat{p}_{i,j}^{(01)}\right)^{(1-c_i)c_j} \left(\hat{p}_{i,j}^{(10)}\right)^{c_i(1-c_j)} \left(\hat{p}_{i,j}^{(11)}\right)^{c_i c_j}. \tag{5}$$

Mithilfe des χ^2-Unabhängigkeitstests auf C_i und C_j untersuchen wir die Abhängigkeiten zwischen Positionen i und j unter der Häufigkeitsverteilung $\hat{P}_{\boldsymbol{C}_{i,j}}$. Zunächst wird die Abhängigkeit der ersten Position $i = 1$ von allen anderen Positionen $j \neq i$ getestet. Dies wird für alle Trägermedien wiederholt. Die dabei gesammelten p-Werte werden je Position j gemittelt und spaltenweise aufgereiht in Abb. 4 (links) dargestellt.

Danach beobachten wir die Abhängigkeiten zwischen allen Positionspaaren $i \neq j$. Dazu wird der mittlere p-Wert für alle j und $i \neq j$ bestimmt, je Position j gemittelt, über alle Trägermedien gemittelt und spaltenweise in Abb. 4 (rechts) dargestellt. Abbildungsposition j stellt dann den gemittelten p-Wert der χ^2-Unabhängigkeitstests über die alle Positionen $i \neq j$ im Vergleich zu Position j dar.

Zum Verständnis der Verteilung der p-Werte, bestimmen wir den Anteil der p-Werte $\leq 5\%$ je Trägermedium. Die Anteile sind in Histogramm Abbildung 5 aggregiert.

Diskussion der Beobachtungen. In Abbildung 4 (links) sind niedrige p-Werte für die Nachbarschaft der ersten Position zu erkennen. Die beobachtete Abhängigkeit lässt sich aus der Konstruktion der Matrix $\mathbb{H}$ (1) erklären, die lineare Abhängigkeiten zwischen nahen Positionen impliziert. Die ersten $h\alpha^{-1}$ Positionen müssen gemeinsam so gewählt werden, dass deren Paritäten mit den ersten h Nachrichten-Bits übereinstimmen. Diese Abhängigkeiten setzen sich kaskadierend über den gesamten Änderungsvektor fort.

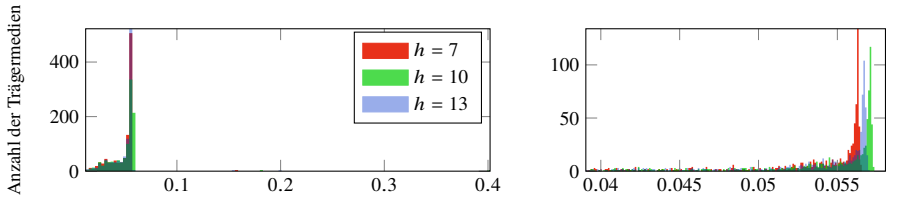


Abb. 5: Histogramme der Anteile von p-Werten $\leq 5\%$. Dabei fließt je Trägermedium der Anteil der p-Werte $\leq 5\%$ über alle Positionen ein. Dargestellt ist ein Histogramm über alle Anteile (links) und ein Histogramm über die Anteile in der Nachbarschaft der Höhepunkte um 0.05 (rechts).

Ein gegenteiliger Effekt ist zum Ende des Änderungsvektors zu beobachten, und das für beide aggregierten Abhängigkeiten in Abbildung 4 (links und rechts): Die letzten Positionen weisen hohe gemittelte p-Werte auf, was bedeutet, dass sie dazu tendieren unabhängig von den restlichen Positionen zu sein. Auch dies kann als Konsequenz der Konstruktion der Matrix $\mathbb{H}$ erklärt werden. Die letzten α^{-1} Positionen des Steganogramms beeinflussen lediglich das letzte Nachrichten-Bit. Die letzten α^{-1} Positionen sind lediglich implizit über die Parität des letzten Nachrichten-Bits mit den vorherigen $h\alpha^{-1}$ Positionen verbunden. Daher ist der kaskadierende Effekt der Auswahl bestimmter Änderungen an diesen Positionen und die damit verbundene Abhängigkeit verhältnismäßig klein.

Diese Untersuchung ist unter Kenntnis der Permutation durchgeführt, sodass die Positionen in der Reihenfolge der Kodierung betrachtet werden können. Im Fall einer unbekannten, aber für ausreichend viele Steganogramme fixen Permutation, würde eine solche Untersuchung potentiell das Rekonstruieren der Permutation ermöglichen und damit deren Beitrag zur steganographischen Sicherheit neutralisieren.

Des Weiteren kann ein ausreichend mächtiger Angreifer bei einer unbekannten aber festen Permutation ein Histogramm der p-Werte, wie in Abbildung 5, erstellen. Der erwartete Anteil von Positions-Paaren mit p-Werten $\leq 5\%$ ist für unabhängig gewählte Änderungen 5%. Aufgrund der Abhängigkeit des Anteils von dem gewählten Kodierungsparameter, ist die Beobachtung von mittleren Anteilen über 5% nicht nur für die Steganalyse relevant, sondern offenbart potenziell auch Informationen über die Permutation.

In unserem Experiment ist der mittlere Anteil der p-Werte $\leq 5\%$ für den Kodierungsparameter $h = 10$ höher als für $h = 13$. Dies bedeutet, dass bei $h = 10$ im Mittel weniger Positionspaare den Unabhängigkeitstest bestehen, als bei $h = 13$. Die Ursache dafür bilden die Werte der Submatrix $\hat{\mathbb{H}}$, da sie maßgeblich dafür sind, wie sich Abhängigkeiten über den Änderungsvektor hinweg bilden. Derartige Statistiken sollten die Wahl einer Submatrix in zukünftigen Forschungsarbeiten anleiten.

6 Abschließende Bemerkungen

Es ist weitere Forschung nötig um den tatsächlichen Verlust an steganographischer Sicherheit zu quantifizieren, der durch die Abhängigkeitsstrukturen in der Kodierung entsteht. Dazu könnte bei der Steganalyse das Wissen naher paarweiser Abhängigkeiten direkt genutzt werden, oder indirekt durch den Versuch, die Permutation wiederherzustellen. Auch eine gründliche Untersuchung von OC-STC und der Vermeidung von negativen Ausreißern der Änderungshäufigkeiten verbleibt im Rahmen künftiger Forschung zu erbringen.

Zusammenfassend hat diese Forschungsarbeit gezeigt, dass die Kodierung als Teil eines steganographischen Systems relevante Fragen offen lässt. Dabei widersprechen die hier präsentierten Ergebnisse nicht unmittelbar den bisherigen Erkenntnissen bezüglich steganographischer Sicherheit, die auf der Simulation von Einbettungen durch die zufällige Wahl optimaler Änderungsvektoren aufbauen. Vielmehr können die Ergebnisse als obere Grenze der Sicherheit von Systemen betrachtet werden, welche die Simulation durch Einbettung echter Nachrichten mit STC ersetzen.

Für weitere Statistiken, Details zu der Durchführung der Experimente, Erklärungen im Kontext von Bildern als Trägermedien und eine Performance-Untersuchung sei auf die englischsprachige Ursprungsfassung [KPB17] verwiesen.

Danksagung

Wir danken Alexander Schlögl für die Hilfe bei der Implementierung der STC auf dem HPC, Pascal Schöttle und den anonymen Reviewern der IWDW und GI Sicherheit für die wertvollen Kommentare.

Die präsentierten empirischen Ergebnisse wurden mithilfe der HPC Infrastruktur “LEO” der Universität Innsbruck erzielt. Diese Forschung wurde von der Archimedes Privatstiftung, Innsbruck und der Deutschen Forschungsgemeinschaft (DFG) unter “Informationstheoretische Schranken digitaler Bildforensik” gefördert.

Literaturverzeichnis

[BFP11] Bas, Patrick; Filler, Tomáš; Pevnỳ, Tomáš: “Break Our Steganographic System”: The Ins and Outs of Organizing BOSS. In (Filler, Tomáš; Pevný, Tomáš; Craver, Scott; Ker, Andrew, Hrsg.): Information Hiding (13th International Conference). Jgg. 6958 in Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, S. 59–70, 2011.

[CSB14] Carnein, Matthias; Schöttle, Pascal; Böhme, Rainer: Predictable Rain? Steganalysis of Public-Key Steganography using Wet Paper Codes. In: ACM Information Hiding and Multimedia Security Workshop. Salzburg, Austria, S. 97–108, 2014.

[FF10a] Filler, Tomáš; Fridrich, Jessica: Gibbs construction in steganography. IEEE Transactions on Information Forensics and Security, 5(4):705–720, 2010.

[FF10b] Filler, Tomáš; Fridrich, Jessica: Minimizing additive distortion functions with non-binary embedding operation in steganography. In: IEEE International Workshop on Information Forensics and Security (WIFS). Tenerife, Spain, S. 1–6, 2010.

[FFJ] Filler, T.; Fridrich, J.; Judas, J.: , Syndrome Trellis Coding, Binghamton reference implementation. http://dde.binghamton.edu/download/syndrome/ (letzter Zugriff: Juni 2017).

[FJF10] Filler, Tomáš; Judas, Jan; Fridrich, Jessica: Minimizing embedding impact in steganography using trellis-coded quantization. In: Proceedings of SPIE-IS&T Electronic Imaging: Security, Forensics, Steganography and Watermarking of Multimedia Contents X. San Jose, CA, S. 754105–754105, 2010.

[FJF11] Filler, Tomáš; Judas, Jan; Fridrich, Jessica: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security, 6(3):920–935, 2011.

[Ke08] Ker, Andrew D: Locating steganographic payload via WS residuals. In: ACM Multimedia and Security Workshop. Oxford, UK, S. 27–32, 2008.

[KPB17] Köhler, Olaf Markus; Pasquini, Cecilia; Böhme, Rainer: On the Statistical Properties of Syndrome Trellis Coding. In (Krätzer, Christian; Shi, Yun-Qing; Dittmann, Jana; Kim, Hyoung-Joong, Hrsg.): Digital Forensics and Watermarking, 16th International Workshop, IWDW 2017. Jgg. 10431 in Lecture Notes in Computer Science, Springer, Berlin Heidelberg, S. 331–346, 2017.

[PK14] Pevnỳ, Tomäš; Ker, Andrew D: Steganographic key leakage through payload metadata. In: ACM Information Hiding and Multimedia Security Workshop. Salzburg, Austria, S. 109–114, 2014.

[RC12] Reed, Irving S; Chen, Xuemin: Error-control coding for data networks. Springer Science & Business Media, New York, US, 2012.

[SCF16] Sedighi, Vahid; Cogranne, Rémi; Fridrich, Jessica: Content-adaptive steganography by minimizing statistical detectability. IEEE Transactions on Information Forensics and Security, 11(2):221–234, 2016.

[Wa93] Wang, Yuan H: On the number of successes in independent trials. Statistica Sinica, 3(2):295–312, 1993.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 95

SDN Ro^{2}tkits: A Case Study of Subverting A Closed Source SDN Controller

Christian Röpke[1]

Abstract: An SDN controller is a core component of the SDN architecture. It is responsible for managing an underlying network while allowing SDN applications to program it as required. Because of this central role, compromising such an SDN controller is of high interest for an attacker. A recently published SDN rootkit has demonstrated, for example, that a malicious SDN application is able to manipulate an entire network while hiding corresponding malicious actions. However, the facts that this attack targeted an open source SDN controller and applied a specific way to subvert this system leaves important questions unanswered: How easy is it to attack closed source SDN controllers in the same way? Can we concentrate on the already presented technique or do we need to consider other attack vectors as well to protect SDN controllers?

In this paper, we elaborate on these research questions and present two new SDN rootkits, both targeting a closed source SDN controller. Similar to previous work, the first one is based on Java reflection. In contrast to known reflection abuses, however, we must develop new techniques as the existing ones can only be adopted in parts. Additionally, we demonstrate by a second SDN rootkit that an attacker is by no means limited to reflection-based attacks. In particular, we abuse aspect-oriented programming capabilities to manipulate core functions of the targeted system. To tackle the security issues raised in this case study, we discuss several countermeasures and give concrete suggestions to improve SDN controller security.

Keywords: Software-defined networking; SDN controller security; SDN rootkits

1 Introduction

Academia and industry have brought forward Software-Defined Networking (SDN) for quite a long time already. The key driver was (and probably is) the OpenFlow protocol [Mc08] which was introduced in 2008. From then on, many studies have been presented which vary from OpenFlow improvements [Opa], numerous SDN controllers [Gu08, Mc, Fl, Opb, ON, Yo17], network simulation software [LHM10], hardware switches with OpenFlow support to various SDN applications [HPa]. In addition, SDN is on the big player's business agenda including network vendors such as Cisco and Juniper, software vendors such as Microsoft and Oracle, network operators such as Verizon and Deutsche Telekom, and data center operators such as Google and Facebook. Major reasons for this are the ability to solve pressing network

[1] Ruhr-Universität Bochum, Lehrstuhl für Systemsicherheit, Universitätsstrasse 150, 44801 Bochum, NRW, christian.roepke@rub.de

doi:10.18420/sicherheit2018_07

problems easier than in the past [FRZ13], the potential to reduce costs [Ho12, Va17], and opportunities to improve network security [Sh16].

In software-defined networks, SDN controllers play a crucial role [Op13, Op14]. They primarily connect network hardware (aka. SDN switches) and network software which is responsible for making forwarding decisions (aka. SDN applications). Programming SDN switches is performed either proactively by inserting flow rules into an SDN switch or reactively by reacting on packets, which are delegated to an SDN controller because of a missing flow rule. For the benefit of SDN applications, SDN controllers provide a global view of the network and allow operating on it in an abstract manner. Obviously, this control system is of high interest for an attacker as compromising it would allow network-wide adverse manipulations. As recent work [RH15b] concentrates on open source SDN controllers, it is still unclear whether closed source systems are affected equally serious. It is also uncertain if simply blocking Java reflection for SDN applications would solve the problem, or if an attacker can implement SDN rootkits using other techniques.

To complement existing work, we present two new SDN rootkits and use them as demonstrators to discuss these open issues. The first one is based on Java reflection which is enabled by default even for closed source SDN controllers. The second one takes advantage of Aspect-Oriented Programming (AOP) which must not be supported by an SDN controller in its default settings. However, we demonstrate how an attacker can add required AOP software via bypassing a corresponding security mechanism (i. e., signature validation). In order to run these attacks, we target the HP VAN SDN controller [HPb] as it is supposed to provide robust security [HPc] while several release generations indicate a certain degree of maturity.

2 Background

2.1 Java Reflection in SDN Rootkits

Java is an object-oriented programming language which includes that access to class internals can be restricted. By choosing an access modifier like *public* or *private*, a developer can specify which other classes can have access to a variable or method of a class. Especially, private variables and methods are not supposed to be accessible by objects of other classes. However, Java also provides a mechanism called reflection [Ora] which allows to bypass this mechanism. The already presented SDN rootkit [RH15b] takes advantage of this mechanism and modifies private fields as well as invokes private methods. In particular, private field manipulation was used for both removing rootkit artifacts from internal inventories and replacing internal services by malicious versions. In the latter case, source code of OpenDaylight was used to re-implement internal services with malicious filtering functions. To replace such an internal service, an object of a modified version was created which was then set to a private variable which was supposed to point to the original service.

In addition, private methods were invoked in order to bypass internal cache updates which are supposed, among others, to keep internal flow rule databases up-to-date. By skipping such updates, the SDN rootkit was able to hide manipulations from these databases, which are used by internal services as well as by SDN applications to monitor network changes.

2.2 Aspect-Oriented Programming

Aspect-oriented programming is a programming paradigm which complements object-oriented programming by enabling cross-cutting concerns [Ki97]. It brings flexibility and is especially capable of improving security which is in fact a cross-cutting concern [DWJP05, PS08, VBC01, DWVDD02]. So-called aspects act similar to classes while containing, among others, so-called advices and pointcuts. An advice specifies the code manipulation and a pointcut defines a point in the program flow where such an advice is supposed to be applied. A typical use case for AOP is adding log functionality, for example, to observe when a certain method is called and what arguments it receives. Instead of modifying all classes which use such a method, AOP enables to define an aspect which modifies all the existing code at once. For example, additional code can be executed before or after a method of interest. In case of Java, code manipulation is typically implemented on the byte code level by weaving manipulations either into already compiled classes (compile-time weaving) or while loading a class (load-time weaving). Thus, AOP for Java allows to modify closed source systems such as SDN controllers, for example, by adding security checks before accessing critical controller functions.

2.3 Attacker Model

Throughout the rest of this paper, we assume an attacker which is able to install a malicious SDN application. This can be achieved in several ways: (i) an attacker can lure administrators into installing such an application, (ii) a security vulnerability can be exploited to bypass the security mechanism taming such applications as a whole [Orb, Orc] or in parts [Orf, Ore, Ord], and (iii) an attacker can steal a certificate in order to correctly sign a malicious application[FMC11]. In addition, we allow this malicious SDN application to use SDN controller services in the same way other SDN applications can do. This includes, for example, reading the network topology (e. g., to look for an interesting target) and re-programming network devices (e. g., to manipulate the network).

3 Subverting the HP Controller

An SDN rootkit faces several challenges with respect to its primary goal, i. e., hiding its existence. First, it is obliged to hide the components which become visible during its

installation. For instance, SDN controllers typically add information to internal inventories such as a list of installed SDN applications and a list of applications, which are allowed to handle network packets. In addition, the HP controller provides a protection mechanism which is supposed to prevent the installation of unwanted SDN applications. In particular, an SDN application's signature is validated during the installation procedure in order to prevent unsigned software.

Second, an SDN rootkit typically wants to hide malicious network manipulations such as added malicious flow rules. As providing the current network state is a core function of SDN controllers, also the HP controller provides a corresponding interface. An attacker must manipulate this service in order to hide the presence of adverse network manipulations, for example, from a monitoring application.

Third, network manipulations typically trigger network events which are observed by SDN controllers. As this can reveal malicious network manipulations, an attacker is obviously interested in hiding corresponding artifacts as well. In case of OpenFlow, for example, *flow_mod* messages are generated when (re)-programming SDN switches and *flow_removed* events are sent to inform SDN controllers about flow rule removals. In order to monitor such network events, the HP controller provides two separate listener services. On the one hand, a so-called *flow listener* allows to monitor flow_mod OpenFlow messages. On the other hand, a so-called *message listener* enables the observation of various messages including flow_mod and flow_removed.

Fourth, we face additional challenges with respect to closed source SDN controllers. On the one hand, it is more difficult to understand the internal functioning of an SDN controller. As this is particularly important to identify interesting spots within an SDN controller, it can complicate subverting critical functions significantly. On the other hand, abusing Java reflection in the way used by previous work [RH15b] does not work anymore. For example, replacing a controller component by implementing a malicious version which is based on the SDN controller's source code (see Section 2.1) is not possible. This reflection-based technique was, however, essential to hide critical SDN rootkit artifacts.

3.1 Abusing Java Reflection

As mentioned before, our first SDN rootkit uses Java reflection to subvert the HP controller. As solving aforementioned challenges varies in difficulty, we describe this rootkit's solutions depending on its difficulty and start with the most difficult one. The most challenging parts are to understand the internal functioning of the HP controller and to find suitable spots, which must be manipulated in order to hide malicious network manipulations. Based on the HP controller's programming guide, its API documentation, and by means of decompiling byte code, an attacker is able to solve these challenges despite the controller's complexity. For example, the documentation says that *getFlowStats()* is responsible for listing installed flow rules and that it is provided by a so-called *ControllerService*, which is implemented

by a class called *ControllerManager.class*. To list interesting private fields of this class (e. g., *FlowTrk flowTrk* and *ListenerService ls*), we use Java reflection. Note that an attacker can also take advantage of a Java decompiler such as Procyon[Mi] to ease this process. According to documentation, a flow tracker is a sub-component of the controller service that is responsible to manage reading and writing of flow rules. In particular, it is involved in case a controller component or an SDN application calls getFlowStats() in order to request a list of currently installed flow rules. The documentation does not say much about a listener service, but it seems that it is responsible for managing a list of message listeners which, for example, can receive OpenFlow multipart response messages. This is indicated by a private variable called *msgHandlers* which contains, among others, the message listener of the aforementioned flow tracker. Such OpenFlow messages are particularly important in this context as they also contain a list of current flow rules. Based on these insights, we abuse Java reflection and subvert critical functions of the HP controller without re-using its source code. Figure 1 illustrates the mechanism we implement to hide malicious flow rules, to fake the existence of removed legitimate flow rules, and to hide adversely modified flow rules.

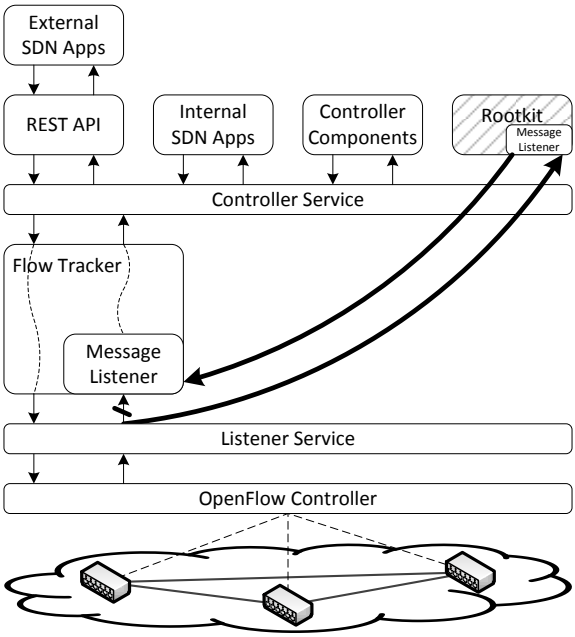


Fig. 1: Hiding network manipulations

In detail, we hook the control flow when getFlowStats() is called as this function is responsible for returning the current network state in terms of installed flow rules. The normal control flow starts when an SDN application (or a controller component) calls this controller service's function. In turn, this service uses its flow tracker sub-component and passes this request towards the network. The flow tracker in turn uses the listener service which in turn uses a component called *openflow service* that sends OpenFlow multipart

request messages to the network. As a result, the corresponding SDN switch returns a multipart response message including all of its flow rules including the malicious ones. Inside the HP controller, receiving such an OpenFlow message generates an event which is passed to the aforementioned flow tracker's message listener. Normally, this flow tracker would pass the event data to the controller service, which returns the requested flow rules to the component which was calling getFlowStats() in the first place. However, we hook into this control flow in order to enable our rootkit to filter out rootkit related data *before* it is passed to the controller service. To achieve this, we implement a message listener which any SDN application is allowed to do. Since the flow tracker's message listener is called before the one of our rootkit, filtering out data is not effective yet. For this purpose, we must manipulate the control flow in a way that the rootkit's message listener is executed before or instead of the flow tracker's one. By applying Java reflection we manipulate the private list of message handlers (i. e., *msgHandlers*), which is managed by the listener service, and replace the flow tracker's message listener object by our rootkit's listener. From now on, all event data, which is supposed to reach the flow tracker's message listener, is in fact passed to our rootkit. In order to keep the HP controller functioning, we also invoke the flow tracker's message listener (i. e., the event function of its message listeners) after we filter out rootkit related data. Thus, we are able to to hide malicious network manipulations from all controller components as well as from all potential SDN security applications.

The remaining challenges (i. e., hiding rookit components and hiding from network events) are solved as follows. In order to hide artifacts generated through a normal installation procedure, we avoid using the controller's web interface, but use the controller's deploy folder *virgo/pickup*. As this folder can be used for hot deployment tasks, the HP controller automatically attempts to install all files which are copied to this folder. When installing our rootkit like this, it neither appears in the list of installed SDN application nor does the HP controller's signature validation mechanism raises any alert. Note that other popular SDN controllers such as OpenDaylight [Opb] and ONOS [ON] also support such a deplay folder. Hiding this bundle from controller internal functions can be achieved as presented in previous work [RH15b]. With respect to hiding from network events, the HP controller provides two valuable services: (i) flow listeners and (ii) message listeners. Both are suitable to monitor critical network messages such as flow_mod and flow_removed. These can be used, for example, by a monitoring application in order to observe what flow rules are actually added, modified, and deleted. By comparing this view of the network with data provided by getFlowStats(), differences in terms of hidden manipulations can be revealed. Unfortunately, there are two drawbacks regarding these components. First, the flow listener function does not work correctly in version 2.7.10 (and probably earlier versions) but only in the newest version (i. e., 2.7.18). Second, registered flow listeners get informed about such critical network events only if a special flow programming function is used (i. e., sendConfirmedFlowMod()). For example, the controller's REST API uses this function which is based on OpenFlow barrier response messages. However, in case we call *sendFlowMod()* no such barrier request messages are generated after sending a flow_mod message. Thus, our rootkit can add, modify, and delete flow rules without notifying a

flow listener. In a similar way, registered message listeners get also notified only in case sendConfirmedFlowMod() is used, which our SDN rootkit does not.

To evaluate this first SDN rootkit, we run several tests. First, we test default monitoring capabilities of the HP controller and potential monitoring capabilities of SDN security applications. Former tests include reviewing the list of SDN applications provided via the controller's web interface, and observing the list of flow rule statistics which is provided by the web interface as well as by the controller's REST API. For the latter tests, we implement an SDN application with comprehensive monitoring capabilities. As a result, we find that our rootkit is capable of hiding its artifacts from all these tests. In addition, we not only test the rootkit against the initially targeted HP controller version (i. e., 2.7.10) but also against other versions (i. e., v2.6.11, v2.7.10, v2.7.16, and the newest version 2.7.18). To attack all these versions, only a few changes are necessary. Regarding v2.6.11, we recompile our rootkit with the SDK shipped with this controller version. With respect to v2.7.18, we change a single letter in the rootkit code as in this controller version a variable's name changes from *ls* to *lm*.

3.2 Abusing Aspect-Oriented Programming

Our second SDN rootkit uses aspect-oriented programming instead of Java reflection. Since by default the HP controller only provides limited AOP support via the Spring AOP framework [Ro], we install and use AspectJ [Xe] which is more powerful. To achieve this, we include AspectJ-related software within the rootkit package. Then, during the installation of this rootkit the HP controller is configured appropriately. In particular, the rootkit extracts AspectJ files to the correct folders, manipulates HP controller configuration files to add AspectJ support, and finally triggers a controller restart. As adding AspectJ support is achieved during runtime, an attacker can used it as this capability were enabled by default. Note that an attacker may find a way to use the AOP support, which is shipped with the HP controller, instead of AspectJ.

Based on insights gained during the implementation of the first rootkit, we solve several challenges in the same way. In particular, finding suitable spots within the HP controller, hiding rootkit artifacts generated during its installation, and hiding from network events are solved in the same fashion as described before (see Section 3.1). The remaining and more interesting challenge is replacing Java reflection by AOP as this would allow an attacker to subvert SDN controllers although Java reflection is prohibited. For this purpose, our second rootkit uses AspectJ to replace entire methods in order to hook the HP controller's normal control flow. Instead of manipulating a list of message listeners via Java reflection, this eases subverting SDN controllers significantly.

Figure 2 shows how this is achieved to hide malicious flow rules. First, we define a *pointcut* to specify a point during the execution of the *getFlowStats* method. Then, we replace the original method by our own one by using an *around* advice. Now, in case OpenFlow

```
pointcut pc(DataPathId d,TableId t): call(* *.getFlowStats(DataPathId,TableId)) && args(d,t);
List<MBodyFlowStats> around(DataPathId d, TableId t) : pcName(d, t) {
  // get actual network state
  List<MBodyFlowStats> orig = proceed(d, t);
  // create filtered data set
  List<MBodyFlowStats> filtered = new ArrayList<MBodyFlowStats>();
  for (int i = 0; i < orig.size(); i++) {
    for (int j = 0; j < flows_to_hide.size(); j++) {
      // if current flow rule should NOT be hidden, add to filtered
    }
  }
  return filtered;
}
```

Fig. 2: AspectJ Example of Hiding Adverse Network Manipulations

messages are received by the HP controller, this replacement can handle them. For the purpose of hiding malicious flow rules, we compare the received list of installed flow rules with a list of flow rules to hide. In case of a hit, we filter it out. Finally, we return a set of filtered flow rules to the caller of getFlowStats(), which is typically an SDN application or a controller component. As a result, our AOP-based rootkit is able to hide malicious network manipulations from the HP controller successfully. In order to demonstrate the effectiveness of this rootkit, we re-run the tests performed to evaluate our first rootkit. This includes reviewing the list of SDN applications and observing the list of flow rule statistics both provided by the controller's web interface as well as by the controller's REST API, and running an SDN application which implements monitoring capabilities.

4 Limitations and Discussion

Since our SDN rootkits take advantage of Java-specific capabilities, these attacks primarily affect Java-based SDN controllers. However, SDN controllers written in other programming languages also suffer from malicious SDN applications [Sh14]. In particular, AOP support is not limited to Java and, thus, can enable attackers to replace critical code, for example, for C++-based SDN controllers [OS]. In addition, our second rootkit currently requires the HP controller's hot deployment mechanism in order to install additional software. As many closed source SDN controllers are based on OpenDaylight [SD15], which also supports such a mechanism, an attacker can abuse this in a similar fashion. Furthermore, we target only one closed source SDN controller as attacking various systems would go beyond the scope of this work. Nevertheless, we argue that several other systems are equally affected as many closed source SDN controllers use OpenDaylight as a basis, which is currently neither able to prevent Java reflection nor AOP operations.

To tackle the raised security concerns, we suggest improvements which are specific to the HP controller, on the one hand, and generally applicable for SDN controllers, on the other hand. With respect to the HP controller, flow and message listeners should be enabled to receive flow programming related events independent of barrier messages. This would

allow a more realistic observation of messages which are related to network manipulations. Furthermore, validating signatures of software should be activated for the entire controller platform. In particular, we strongly recommend to cover the controller's hot deployment folder *virgo/pickup*. Moreover, we suggest to ship the controller with default monitoring capabilities which is capable of finding obvious inconsistencies. In addition, a mechanism should be provided which protects critical controller mechanisms from being hijacked. Particularly, invoking message listeners must be protected in a way that listeners of SDN applications are not processed before the ones of controller components.

In more general, SDN controller vendors may be interested in implementing security improvements such as (i) putting SDN applications into a sandbox, (ii) tracking reflection and AOP related critical operations, and (iii) comparing the actual network state with the state provided to SDN applications. In particular, several sandbox systems have been proposed already [RH15a, Sh14, CTB16, Yo17]. With such a system, our Java reflection and AOP based attacks can be prevented, for example, by denying access to corresponding critical Java operations. However, it is worth noting that the use of such operations is not malicious per se and, thus, SDN applications can use them in a benign manner as well. Hence, we suggest to track corresponding critical operations in order to prevent only a malicious utilization. Another possibility is to compare the actual network state provided by network devices with the network view provided to SDN applications. Hereby, discrepancies such as hidden flow rules can be easily revealed by a dual-view comparison [Ta17].

5 Related Work

The problem of malicious SDN applications was first raised by Porras et al. [Po12]. The authors presented a new technique which enables attackers to bypass existent flow rules by exploiting the standard OpenFlow instructions *set* and *goto*. Shin et al. [Sh14] and Röpke et al. [RH15a, RH16] extended this work by presenting SDN applications which can harm SDN controllers by implementing rudimentary malicious logic. For this purpose, the authors exploited the fact that many SDN controllers run their SDN applications within the same execution environment. In addition, Röpke et al. [RH15b] demonstrated a more sophisticated malicious SDN application which is able to compromise an SDN controller and, thus, an entire SDN via abusing Java reflection. Complementary to aforementioned research, we adopt Java reflection-based attacks to a closed source SDN controller and, additionally, present an AOP based technique to compromise SDN controllers. Although manipulating Java programs via AOP is not new in general [Wi09], abusing AspectJ to subvert SDN controllers is new in the context of SDN.

6 Conclusion

In this paper, we present two new SDN rootkits which aim to subvert a closed source SDN controller. The first one extends existing Java reflection based attacks while the

second one abuses aspect-oriented programming techniques to subvert critical controller functions. Both rootkits are able to compromise the HP controller in a default setup and both attacks do subvert this controller on such a deep level that even multiple release versions are affected. Clearly, this shows that an SDN rootkit is not only a severe threat to open source SDN controllers but also to closed source ones. We also demonstrate that preventing SDN applications from accessing reflection-related operations is not enough to protect SDN controllers. Further attack vectors must be considered. Moreover, we find that using Java reflection to subvert SDN controllers heavily depends on a concrete implementation. Thus, compromising another open or closed source SDN controller probably raises new challenges with respect to finding suitable spots for hooking the control flow. To prevent such attacks, we finally discuss several countermeasures including concrete suggestions to improve security of the targeted SDN controllers.

References

[CTB16] Chandrasekaran, Balakrishnan; Tschaen, Brendan; Benson, Theophilus: Isolating and Tolerating SDN Application Failures with LegoSDN. In: ACM Symposium on SDN Research. 2016.

[DWJP05] De Win, Bart; Joosen, Wouter; Piessens, Frank: Developing Secure Applications through Aspect-Oriented Programming. Aspect-Oriented Software Development, 2005.

[DWVDD02] De Win, Bart; Vanhaute, Bart; De Decker, Bart: Security through Aspect-Oriented Programming. Advances in Network and Distributed Systems Security, 2002.

[Fl] Floodlight. `www.projectfloodlight.org/floodlight/`, Accessed: 2018-02-13.

[FMC11] Falliere, Nicolas; Murchu, Liam O; Chien, Eric: W32. stuxnet dossier. White paper, Symantec, 2011.

[FRZ13] Feamster, Nick; Rexford, Jennifer; Zegura, Ellen: The Road to SDN. ACM Queue: Tomorrow's Computing Today, 2013.

[Gu08] Gude, Natasha; Koponen, Teemu; Pettit, Justin; Pfaff, Ben; Casado, Martín; McKeown, Nick; Shenker, Scott: NOX: Towards an Operating System for Networks. ACM SIGCOMM Computer Communication Review, 2008.

[Ho12] Hoelzle, Urs: OpenFlow @ Google. Open Networking Summit, 2012.

[HPa] HP Open Ecosystem Breaks Down Barriers to Software-Defined Networking. `www.hp.com`, Accessed: 2018-02-13.

[HPb] HP VAN SDN Controller. `www.hp.com`, Accessed: 2018-02-13.

[HPc] HP Virtual Application Networks SDN Controller: The building block of HP SDN ecosystem. `www.hp.com`, Accessed: 2018-02-13.

[Ki97] Kiczales, Gregor; Lamping, John; Mendhekar, Anurag; Maeda, Chris; Lopes, Cristina; Loingtier, Jean-Marc; Irwin, John: Aspect-Oriented Programming. In: European conference on object-oriented programming. 1997.

[LHM10] Lantz, Bob; Heller, Brandon; McKeown, Nick: A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. In: ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. 2010.

[Mc] POX SDN Controller. github.com/noxrepo/pox/, Accessed: 2018-02-13.

[Mc08] McKeown, Nick; Anderson, Tom; Balakrishnan, Hari; Parulkar, Guru; Peterson, Larry; Rexford, Jennifer; Shenker, Scott; Turner, Jonathan: OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer Communication Review, 2008.

[Mi] Procyon. bitbucket.org/mstrobel/procyon/, Accessed: 2018-02-13.

[ON] Open Network Operating System. onosproject.org, Accessed: 2018-02-13.

[Opa] OpenFlow Switch Specification. www.opennetworking.org, Accessed: 2018-02-13.

[Opb] OpenDaylight Project. www.opendaylight.org, Accessed: 2018-02-13.

[Op13] Software-Defined Networking: The New Norm for Networks. www.opennetworking.org.

[Op14] SDN Architecture Overview (ONF TR-504). www.opennetworking.org.

[Ora] Java Reflection API. docs.oracle.com/javase/7/docs/technotes/guides/reflection/, Accessed: 2018-02-13.

[Orb] Security Alert for CVE-2012-4681. www.oracle.com, Accessed: 2018-02-13.

[Orc] Security Alert for CVE-2013-0422. www.oracle.com, Accessed: 2018-02-13.

[Ord] Sun Alert 1000148.1. download.oracle.com/sunalerts/, Accessed: 2018-02-13.

[Ore] Sun Alert 1000560.1. download.oracle.com/sunalerts/, Accessed: 2018-02-13.

[Orf] Sun Alert 1000975.1. download.oracle.com/sunalerts/, Accessed: 2018-02-13.

[OS] AspectC++. www.aspectc.org, Accessed: 2018-02-13.

[Po12] Porras, Philip; Shin, Seungwon; Yegneswaran, Vinod; Fong, Martin; Tyson, Mabry; Gu, Guofei: A Security Enforcement Kernel for OpenFlow Networks. In: ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. 2012.

[PS08] Phung, Phu H; Sands, David: Security policy enforcement in the OSGi framework using aspect-oriented programming. In: 2008 32nd Annual IEEE International Computer Software and Applications Conference. 2008.

[RH15a] Röpke, Christian; Holz, Thorsten: Retaining Control Over SDN Network Services. In: International Conference on Networked Systems. 2015.

[RH15b] Röpke, Christian; Holz, Thorsten: SDN Rootkits: Subverting Network Operating Systems of Software-Defined Networks. In: Symposium on Recent Advances in Intrusion Detection. 2015.

[RH16] Röpke, Christian; Holz, Thorsten: On Network Operating System Security. International Journal of Network Management, 2016.

[Ro] Spring Framework Reference Documentation. docs.spring.io/spring/docs/current/spring-framework-reference/html/, Accessed: 2018-02-13.

[SD15] SDxCentral: SDN Controllers Report. www.sdxcentral.com, 2015.

[Sh14] Shin, Seungwon; Song, Yongjoo; Lee, Taekyung; Lee, Sangho; Chung, Jaewoong; Porras, Phillip; Yegneswaran, Vinod; Noh, Jiseong; Kang, Brent Byunghoon: Rosemary: A Robust, Secure, and High-Performance Network Operating System. In: ACM SIGSAC Conference on Computer and Communications Security. 2014.

[Sh16] Shin, Seungwon; Xu, Lei; Hong, Sungmin; Gu, Guofei: Enhancing Network Security through Software Defined Networking (SDN). In: Proceedings of the 25th International Conference on Computer Communication and Networks. 2016.

[Ta17] Tatang, Dennis; Quinkert, Florian; Frank, Joel; Röpke, Christian; Holz, Thorsten: SDN-Guard: Protecting SDN controllers against SDN rootkits. In: Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on. 2017.

[Va17] Vahdat, Amin: Cloud Native Networking. Open Networking Summit, 2017.

[VBC01] Viega, John; Bloch, JT; Chandra, Pravir: Applying Aspect-Oriented Programming to Security. Cutter IT Journal, 2001.

[Wi09] Williams, Jeff: Enterprise Java Rootkits: Hardly anyone watches the developers. BlackHat USA, 2009.

[Xe] The AspectJ Programming Guide. eclipse.org/aspectj/doc/next/progguide/index.html, Accessed: 2018-02-13.

[Yo17] Yoon, Changhoon; Shin, Seungwon; Porras, Phillip; Yegneswaran, Vinod; Kang, Heedo; Fong, Martin; O'Connor, Brian; Vachuska, Thomas: A Security-mode For Carrier-grade Sdn Controllers. In: Anual Computer Security Applications Conference. 2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 107

Source Code Patterns of Buffer Overflow Vulnerabilities in Firefox

Felix Schuckert [1 2] felix.schuckert@htwg-konstanz.de
Max Hildner [1] maxhildner@fastmail.com
Basel Katt [2] basel.katt@ntnu.no
Hanno Langweg [1 2] hanno.langweg@htwg-konstanz.de

Abstract: We investigated 50 randomly selected buffer overflow vulnerabilities in Firefox. The source code of these vulnerabilities and the corresponding patches were manually reviewed and patterns were identified. Our main contribution are taxonomies of errors, sinks and fixes seen from a developer's point of view. The results are compared to the CWE taxonomy with an emphasis on vulnerability details. Additionally, some ideas are presented on how the taxonomy could be used to improve the software security education.

Keywords: Buffer Overflow, Source Code Patterns, Vulnerabilities, Code Analysis

1 Introduction

The Common Weakness Enumeration (CWE) [Co17b] top 25 show buffer overflow vulnerabilities (CWE-120) in third place. Buffer overflows have existed for a long time. To discover the reason why buffer overflows still occur in code, we investigated source code samples from the open source web browser Firefox [Fi17]. Different categories for buffer overflow vulnerabilities already exist in CWE. These categories take a technical point of view; they look at aspects such as which memory locations are involved. For example, there are categories for accessing memory on the stack or on the heap. Such categories do not help software developers to avoid buffer overflow vulnerabilities. Developers have to know how vulnerabilities occur and what kind of source code patterns are common for vulnerabilities. Additionally, developers have to know how vulnerabilities can be mitigated. For example, it is important to check inputs carefully and to not misuse memory-critical functions as *memcpy()* that is listed as on of security development lifecycle banned function calls from Microsoft [Mi17]. To fill the gap in the current categorization approaches and provide a structured body of knowledge for software developers to mitigate buffer overflow

[1] HTWG Konstanz, Department of Computer Science, Alfred-Wachtel-Straße 8, 78462 Konstanz, Germany
[2] Department of Information Security and Communication Technology, Faculty of Information Technology and Electrical Engineering, NTNU, Norwegian University of Science and Technology, Teknologivegen 22, 2815 Gjøvik, Norway

doi:10.18420/sicherheit2018_08

vulnerabilities, we reviewed 50 source code samples of buffer overflow vulnerabilities in Firefox. In our review, we considered which types of errors the developers made, which sinks were involved in buffer overflow vulnerabilities and how developers patched the vulnerability. Categories were created based on the results from the reviews. These results are compared to the categories from CWE to see the difference from a developer's point of view.

This paper begins with an overview of related work in section 2. The following section explains how the source code was obtained as well as the review method. The categories for buffer overflows are presented in sections 4, 5 and 6. The last two sections provide a discussion of the results and suggestions for future work.

2 Previous and related work

SQL injection vulnerabilities in 50 source code samples from open source projects were analysed by [SKL17] using a similar method. The reviewed programming language was PHP. Classifications of source code patterns exist. Classifying source code into bad code, clean code and ambiguous code was done by Lerthathairat; Prompoon [LP11]. Metrics in source code like comments, the size of the function, et cetera. were analysed using fuzzy logic to determine which category the source code belongs to. Bad and ambiguous code are considered for refactoring. More security-related work is by Hui et al. [Hu10], using a software security taxonomy for software security tests. The security defects taxonomy was created based on the top 10 software security errors from authoritative vulnerability enumerations. It is categorized into into *induced causes*, *modification methods* and *reverse use methods*. Hui et al. [Hu10] suggested to use their taxonomy as security test cases.

Massacci; Nguyen [MN10] investigated different data sources for vulnerabilities, e.g. Common Vulnerabilities and Exposures (CVE) [Co17a], National Vulnerability Database (NVD) [Na17], et cetera. They checked which data sources were used by other research projects. In their work, Firefox was used as database for the analysis. Semantic templates were derived from the existing CWE database and are supposed to help understand security vulnerabilities by Wu et al. [WSG11]. The authors did an empirical study to prove that these semantic templates have a positive impact on the time until a vulnerability is completely found.

The work by Bishop et al. [Bi12] [Bi10] presents a taxonomy of how buffer overflow vulnerabilities occur, considering which preconditions are required to exploit a vulnerability. These preconditions are not suited to teach software developers to mitigate vulnerabilities. For example, taking into consideration the category that the program can jump to a memory location in the stack. This is relevant for exploiting the vulnerability, but it will not help to understand what kind of mistakes were done in developing the source code. Kratkiewicz; Lippmann [KL05] used a taxonomy of buffer overflow vulnerabilities to create a data set of 291 small C programs. The data set was analysed with static and dynamic code analysis

tools. The tools were then evaluated regarding their detection rate, false positive rate, et ceterea. Ye et al. [Ye16] analysed 100 buffer overflow vulnerabilities and the corresponding patches, using the data to evaluate static code analysis tools. The evaluated tools were *Fortify*, *Checkmarx* and *Splint*. Shahriar; Haddad [SH13] showed how to automatically patch buffer overflow vulnerabilities, including a classification of different types of buffer overflow vulnerabilities. For each of these categories, rules were offered to mitigate the vulnerability. The SEI CERT coding standards [St16] provide an overview on how to implement memory-critical parts in C. The standards are presented as necessary to ensure safety, reliability and security. Non-compliant and compliant code examples help developers to better understand the coding standards.

3 Method

To create the source code pattern categories, selected data sets are needed for the review process. We focus on vulnerabilities which are tracked in the CVE database. We chose source code samples from Firefox because it has many reported buffer overflow vulnerabilities - on average about 30 buffer overflow vulnerabilities per year. Additionally, the Bugzilla [Bu17] platform offers a public discussion about the bug fixes, which helps to identify the relevant source code pattern for the vulnerability. 187 CVE reports are connected to buffer overflow vulnerabilities and Firefox in the time frame from 2010 to 2015 (six years). We choose 2015 as a cut-off to ensure we would have access to a patch as well. We use 50 randomly selected CVE reports which also provide a patch to fix the vulnerability. The patch is determined by a *CONFIRM* flag in the CVE report which indicates the correct Bugzilla report. The bug report contains a link to the patch which fixes the vulnerability. Firefox patches are managed with a version control tool. For each of theses patches, the hash value of the parent version is specified. That version is used as a source code sample containing the buffer overflow vulnerability.

The vulnerable version was reviewed regarding the types of errors made by developers and which sinks were used. A sink is the last instance where unchecked user input can exploit a vulnerability. For example, the function *memcpy()* is a common sink for buffer overflow vulnerabilities. In order to see which errors were made and which sinks were used, a data flow analysis was performed. This was done manually because within the bounds of our project, we could not find a proper tool that was able to analyse such a huge project like Firefox. It is possible that types of errors appear several times because a combinations of errors can also result in a vulnerability. The errors were considered from a developer's point of view. However, the sink is more focused on which critical functions and source code parts are used. This helps developers to recognize critical functions. The patch was used to see how developers fixed the vulnerability. The review of the patch was used to create categories for the fixes.

4 Types of errors

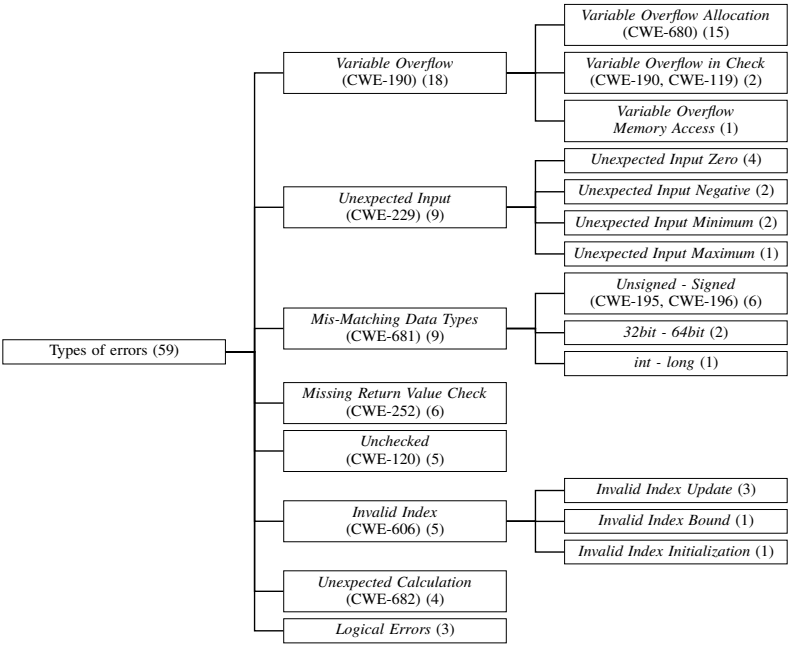


Fig. 1: Taxonomy of errors developers introduced based on the data set.

Figure 1 shows an overview of the categories for the types of errors found in our data set. It has more than 50 assignments because one type of error can lead to other types of errors which then result in a buffer overflow vulnerability. The categories created for the data are the following:

1. ***Variable Overflow***: Many instances of buffer overflows in our review are correlated to integer overflows or underflows. These over-/under-flowed variables are used to check the input size (*Variable Overflow in Check*). Because of the wrong value, these checks pass inputs resulting in a buffer overflow condition. This type of error can be represented in a combination of the following CWE ids: The CWE-190 (Integer Overflow or Wraparound) connected with the keyword *CanProcede* to CWE-119 (Improper Restriction of Operations within the Bounds of a Memory Buffer).

   An overflowed or underflowed variable is used for allocating memory (*Variable Overflow Allocation*). The allocated memory is smaller than the input copied into it. This results in a buffer overflow. The related CWE id is CWE-680, which states that a calculation result is used to allocate memory and an integer overflow causes less memory to be allocated. The allocation of an insufficient amount of memory in our data set occurred in the following sub-patterns:

a) *Allocation too small*: An integer overflow can either have a negative result (signed int) or very small result (unsigned int). These integer overflows occur because user data is included in a calculation. This can be a simple addition to a static value or it can be methods computing a length. As an example, the length of the user input could be the sum of multiple user inputs. If memory is allocated from an integer overflow result, the later usage of the memory will result in a buffer overflow vulnerability.

b) *Existing buffer size check*: Some data sets used already existing buffers and checked if the buffer size had to be increased. An integer overflow in such a check also results in a buffer that is too small.

2. ***Unchecked***: The review shows vulnerabilities where user input reaches methods that are vulnerable for buffer overflows. Source code samples without any checks fall into this category. The corresponding CWE id (CWE-120) explains it as follows: "The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow." Accordingly, this is the classic buffer overflow where input is not checked and then is able to reach critical functions like *memcpy()*.

3. ***Unexpected Input***: This category covers unexpected user inputs. Usually, all of the error types could fall into this category, but it covers inputs that the developers did not expect to occur. For example, the parameter of a method is the size of a file. Accordingly, the parameter must not be negative (*Unexpected Input Negative*). Another example would be a parameter that has a minimum size, thus, falls into the category *Unexpected Input Minimum*. Nevertheless, the parameter can be outside of the expected range because of some other preconditions or special inputs. For example, a specially crafted file would return a negative result as the content length. If a developer uses such premises for memory-critical parts, a buffer overflow vulnerability could occur. One sample also had expected a maximum input (*Unexpected Input Maximum*) of a value. This vulnerability was related to shaders programs which are programs running on the graphics processor. Developers did not think that the value of the input could be higher than the number of existing shaders. CWE-229 (Improper Handling of Values) is best suited to our *Unexpected Input* category because the inputs are not handled properly. The CWE category covers multiple variants like missing values or undefined values. It does not cover numerical values which are too small, too high or in an unexpected range.

4. ***Mis-Matching Data Types***: This category covers vulnerabilities where values of different data types are assigned to each other which is presented in CWE-681 (Incorrect Conversion between Numeric Types). A common example is the assignment from *unsigned int* to *signed int*. These assignments are also covered by CWE-119 (Signed to Unsigned Conversion Error) and CWE-196 (Unsigned to Signed Conversion Error). This type of error occurred in our data set in combination with *Unexpected Calculation* or just as a simple conversion with the outcome of a buffer overflow vulnerability. Also, some samples contain assignments of different variable lengths,

for example, assignments between 32 bit and 64 bit variables. C/C++ does allow the assignment of variables with different data types. It will interpret the bits according to the new variable type. For example, if a negative value is assigned to an unsigned variable, the first bit will be interpreted as the highest value bit. If such an interpretation is unwanted, subsequent checks and the usage of the variable will be problematic. In our sample, this results in buffer overflow vulnerabilities.

5. ***Missing Return Value Check***: These are vulnerabilities where developers do not check the return value. In our data set it was common that the return value of a memory allocation function was not checked. If the allocation is not possible, the allocation functions returns an error code. If the return value is ignored, the pointer will point to a random memory address. Using this pointer to access memory will likely result in a buffer overflow vulnerability. Usually such a situation only happens when the system or program is out of memory. CWE-252 (Unchecked Return Value) is the related category in the CWE list.
6. ***Invalid Index***: These error types include the usage of an invalid index for a loop. It is split into three subcategories. The first is the *Invalid Index Bound* where the bound is invalid. This can happen because of previous errors like an *Unexpected Calculation*. Samples where such a bound is invalid and the index is used to access memory are counted in this category. Another error is that developers did not update the index correctly (*Invalid Index Update*) which also results in a buffer overflow. One sample had an index initialized to an invalid value (*Invalid Index Initialization*). The best fitting CWE category is CWE-606 (Unchecked Input for Loop Condition) because the *Invalid Index* category is related to loops.
7. ***Unexpected Calculation***: This category covers source code samples where unexpected results are obtained during calculation. All the samples had a negative result. The developers did not expect the result to be negative and the values were used in memory-unsafe functions. Another example is assigning a negative result to an unsigned integer. The unsigned integer will interpret the highest bit which is a 1 as a very large value because it was negative when it was represented in a signed datatype. Such an example is represented in CWE ids with the following: CWE-682 (Incorrect Calculation) connected with the keyword *CanFollow* to CWE-681 (Incorrect Conversion between Numeric Types).
8. ***Logical Errors***: Two vulnerable samples showed developers made logical errors. For example, not enough memory was allocated regardless of the input and the following code did write into unintended memory parts. Another sample had an issue where the length of a variable was not updated correctly and that length was used in memory-critical parts. Three samples showed buffer overflow conditions because they had logical errors.

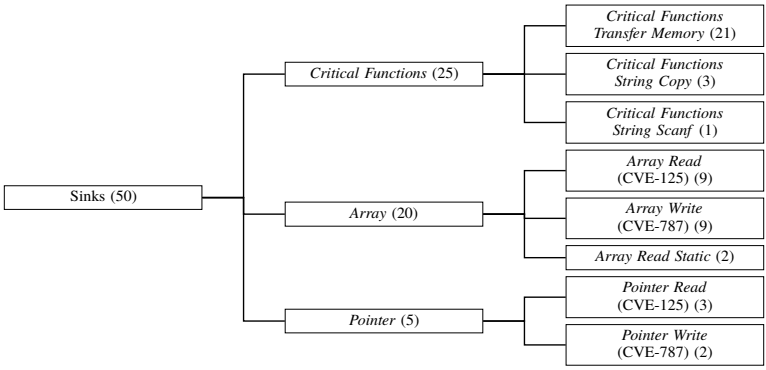


Fig. 2: Taxonomy of sinks based on the data set.

5 Sink categories

What kind of sinks were used in the data set are shown in figure 2. These are classified into the following categories:

1. ***Critical Functions***: Sinks of this category are memory-critical functions. Common functions in C/C++ are *memcpy()* or *memset()*. These functions are categorized into the subcategory *transfer memory*. Three sinks of the data set used a string copy function (*strcpy()*) and one sample used the *scanf()* function. These are functions which are also found in the banned functions list for security development lifecycle [Mi17].
2. ***Array***: Arrays in C/C++ are very similar to pointers. The memory for an array is arranged such that all entries are next to each other. If an array field is accessed using an invalid index, a buffer overflow vulnerability exists. All data sets where the sinks are arrays fall into this category. This can be split into write (CWE-787: Out-of-bounds write) and read (CWE-125: Out-of-bounds read). Two samples performed a read access with a static index. Both of them used the index zero, which is typically used to get the first element of an array.
3. ***Pointer***: The last category for sinks is the misuse of pointers. These are sinks where pointers are used to access memory. This category can be mapped to the CWE-468 (Incorrect Pointer Scaling) category. This category can also be split into subcategories of read (CWE-125: Out-of-bounds read) and write (CWE-787: Out-of-bounds write).

6 Fix categories

The results for the different fixes are categorized and seen in figure 3. They are connected to the different problem types. Fixes are categorized as follows:

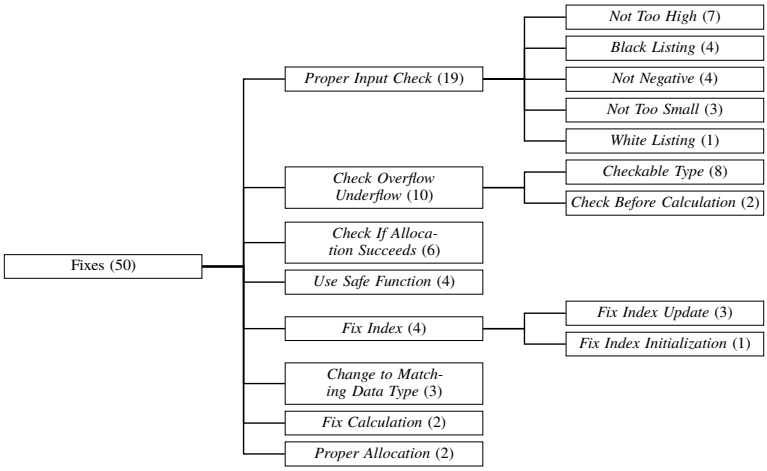


Fig. 3: Taxonomy of fixes for the vulnerabilities based on the data set.

1. ***Proper Input Check***: Fixes for this category are input checks which were completely missing (*Unchecked*). Also fixes which check inputs that developers didn't have in mind fall into this category (*Unexpected Input*). The subcategories are for the different kinds of checks. For example, negative values are a common input developers did not expect. Also some vulnerabilities which have *Variable Overflow in Check* and *Variable Overflow Allocation* as error categories were fixed by checking if the input value was not too high. Also some fixes did just check if a value was not too small. This is commonly a fix when developers thought the input could not be that small. Black listing where specific inputs are filtered out and white listing where only specific inputs are allowed were found as fixes in the data set.
2. ***Check Overflow Underflow***: Firefox has a *checkedint* class which allows to check if an overflow or underflow occurred. Accordingly, fixes used these classes instead of *int* variables and checked for over- and underflow occurrences. Two fixes did check the input, i.e., if an integer overflow occurred in the calculation before using it. This fixes problems from the *Variable Overflow in Check* and *Variable Overflow Allocation* categories.
3. ***Check If Allocation Succeeds***: Some vulnerabilities fall in the error category *Missing Return Value Check*. In our data set, these missing return value checks are related to allocating memory. As the name already hints, fixes in this category check these return values and change the control path accordingly.
4. ***Use Safe Function***: Memory related functions provide a secure function which requires an additional parameter. This parameter is used to restrict the size which is used in the memory-critical function. A common example is *strcpy* and *strcpy_s*. The additional parameter is used to provide the size of the string. This prevents a vulnerability where the source string has no null character or the size of the source

string is larger than the size of the destination string. Four fixes used such functions to remove the vulnerability.

5. ***Fix Index***: This category is related to the error type *Invalid Index*. These errors were fixed by using valid indexes. One instance was fixed by changing the data type so that the index was not invalid any more. The fixes are split into the same subcategories as the error type. For example, one index update fix was implemented by inserting a *break* statement.
6. ***Change to Matching Data Type***: Three vulnerabilities were patched by changing the data type. This fix is related to the *Mis-Matching Data Types* error type category. Three of these kind of errors were patched by changing the data type. The remaining samples were fixed by correcting a previous error which then only resulted in a vulnerability because of mis-matching data types. For example, an integer overflow was fixed, which resulted in a negative result that was assigned to an unsigned integer variable. As long as the value was positive, this did not create a problem.
7. ***Fix Calculation*** Two samples were patched by fixing the calculation. The calculation was adjusted accordingly such that the undesired results will no longer occur.
8. ***Proper Allocation*** The last category of fixes are patches where the allocations were fixed. For example, the allocation did not reserve enough memory. If the allocation was changed such that it allocated the right amount of memory, it falls into this category. Two samples patched the vulnerability by correctly allocating memory.

7 Discussion

Firefox is a well-known open source product and the source code is reviewed a lot. Accordingly, the vulnerabilities from Firefox usually had input checks before potentially dangerous functions or memory accesses were used. The vulnerabilities most often existed because an integer overflow or underflow occurred. It is important to teach developers the right use of variables which may overflow/underflow. Also the assignment of variables with different data types in *C/C++* is problematic and should be avoided. Nevertheless, if these assignments are required, they should be used carefully.

The error types were related to existing CWE categories. CWE-888 contains software fault pattern clusters. The containing category CWE-890 (SFP Primary Cluster: Memory Access) is related to buffer overflow vulnerabilities. This category also has the following subcategories:

- CWE-970 SFP 2. Cluster: Faulty Buffer Access: **covered**
- CWE-971 SFP 2. Cluster: Faulty Pointer Use: **did not occur in data set**
- CWE-972 SFP 2. Cluster: Faulty String Expansion: **did not occur in data set**
- CWE-973 SFP 2. Cluster: Improper NULL Termination: **did not occur in data set**
- CWE-974 SFP 2. Cluster: Incorrect Buffer Length Computation: **covered**

CWE-970 and CWE-974 are covered by our data set. Surprisingly, CWE-971 did not occur in our data set. This is due to the fact that this category has only very specific CWE subcategories, for example, when using a *null* pointer or using pointers to determine a length. Also CWE-972 and CWE-973 did not occur in our data set. There was no vulnerability sample related to an improper *null* termination of a *String* variable. Another related cluster is CWE-969 (SFP Secondary Cluster: Faulty Memory Release) which covers vulnerabilities where memory is released and still used later on. This includes vulnerabilities like "double free" or releasing memory which is not on the heap. Unfortunately, our data set did not cover vulnerabilities which fit into this cluster.

As already stated, Microsoft released a list of banned functions for the security development lifecycle [Mi17]. Most of our sinks that fall into the category *Critical Functions* are found on the list. Our data set contained the critical functions *memmove()* and *memset()*, which are not found in the banned list because these functions are using a restricting length parameter. Only four of samples using a banned function were fixed by using a safe function. 17 of the sinks in our data set did use the function *memcpy()*. According to the list, the function *memcpy_s()* should be used which requires an additional parameter defining the size of destination. None of the patches used the function to fix the vulnerability. It is easy to tell developers to avoid buffer overflow vulnerabilities, but there is a huge list of critical functions. Developers have to know which functions are critical. Static code analysis tools might be useful to find these functions. Nevertheless, in our data set only half of the vulnerabilities use critical functions. Additionally, there are many different permutations of buffer overflow vulnerabilities which makes the mitigation for developers problematic.

Our results show that buffer overflow vulnerabilities are not simply avoided by having a list of critical functions. Buffer overflow vulnerabilities occur in many different permutations and in combination of errors. Accordingly they are not easy to prevent by just learning simple vulnerabilities. Our results provide an overview of source code patterns which are found in our data set. These can be used to teach developers that all kinds of permutations of our categories can result in a buffer overflow vulnerabilities.

8 Conclusions and future work

To minimize the occurrence of buffer overflow vulnerabilities, different source code patterns have to be detected and avoided. To gain a better understanding of how such patterns look like, we analysed 50 buffer overflow CVE reports related to Firefox. We created categories for the types of errors the developers made, what kind of sinks were used and how the developers fixed the vulnerability. These categories were compared to existing CWE categories. Some categories are not found as a direct CWE category. Likewise, our data set does not include all CWE categories. The focus of the categories is seen from a developer's point of view instead of a technical representation of the vulnerability details. This helps to use the categories to teach developers which source code patterns and errors are common for buffer overflow vulnerabilities.

Our patterns could be used to create different learning exercises using different permutations. An interesting point will be to create these exercises automatically. The LAVA tool [Do16] injects buffer overflow vulnerabilities in C code. It would be interesting to integrate our patterns into this tool. This will be an important step because malicious developers might already have developed such tools. It would reveal some limitations and maybe risks which might occur by automatically creating vulnerabilities in the future. Our earlier work [Sc16] is a tool which injects SQL injection vulnerabilities in Java source code using an abstract syntax tree. A similar approach might be possible to inject buffer overflow vulnerabilities in C/C++ code. Another avenue of research would be using these categories to benchmark static code analysis tools. Data sets could be created using different permutations of our categories. It will be interesting to see if all permutations are detected by static code analysis tools as well as the false positives and the false negatives rates of the tools.

References

[Bi10] Bishop, M.; Howard, D.; Engle, S.; Whalen, S.: A taxonomy of buffer overflow preconditions. In. 2010.

[Bi12] Bishop, M.; Engle, S.; Howard, D.; Whalen, S.: A taxonomy of buffer overflow characteristics. In. Vol. 9, pp. 305–317, 2012.

[Bu17] Bugzilla, 2017, URL: https://bugzilla.mozilla.org.

[Co17a] Common Vulnerabilities and Exposures, 2017, URL: https://cve.mitre.org/.

[Co17b] Common Weakness Enumeration, 2017, URL: https://cwe.mitre.org/.

[Do16] Dolan-Gavitt, B.; Hulin, P.; Kirda, E.; Leek, T.; Mambretti, A.; Robertson, W.; Ulrich, F.; Whelan, R.: Lava: Large-scale automated vulnerability addition. In: Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, pp. 110–121, 2016.

[Fi17] Firefox, 2017, URL: https://www.mozilla.org/de/firefox/.

[Hu10] Hui, Z.; Huang, S.; Hu, B.; Ren, Z.: A taxonomy of software security defects for SST. In. Pp. 99–103, 2010.

[KL05] Kratkiewicz, K.; Lippmann, R.: A taxonomy of buffer overflows for evaluating static and dynamic software testing tools. In: Proceedings of Workshop on Software Security Assurance Tools, Techniques, and Metrics. Pp. 500–265, 2005.

[LP11] Lerthathairat, P.; Prompoon, N.: An approach for source code classification to enhance maintainability. In. Pp. 319–324, 2011.

[Mi17] Microsoft: Security Development Lifecycle (SDL) Banned Function Calls, 2017, URL: https://msdn.microsoft.com/en-us/library/bb288454.aspx.

[MN10] Massacci, F.; Nguyen, V. H.: Which is the right source for vulnerability studies?: An empirical analysis on Mozilla Firefox. In. 4:1–4:8, 2010, ISBN: 978-1-4503-0340-8.

[Na17] National Vulnerability Database, 2017, URL: https://nvd.nist.gov/.

[Sc16] Schuckert, F.: PT: Generating Security Vulnerabilities in Source Code. In: Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit. Pp. 177–182, 2016.

[SH13] Shahriar, H.; Haddad, H. M.: Rule-based source level patching of buffer overflow vulnerabilities. In. Pp. 627–632, 2013.

[SKL17] Schuckert, F.; Katt, B.; Langweg, H.: Source Code Patterns of SQL Injection Vulnerabilities. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ACM, 72:1–72:7, 2017.

[St16] Standard, C. C.: SEI CERT. In. 2016.

[WSG11] Wu, Y.; Siy, H.; Gandhi, R.: Empirical results on the study of software vulnerabilities. In. Pp. 964–967, 2011.

[Ye16] Ye, T.; Zhang, L.; Wang, L.; Li, X.: An Empirical Study on Detecting and Fixing Buffer Overflow Bugs. In. Pp. 91–101, 2016.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 119

Is MathML dangerous?

Christopher Späth [1]

Abstract: HTML5 forms the basis for modern web development and merges different standards. One of these standards is MathML. It is used to express and display mathematical statements. However, with more standards being natively integrated into HTML5 the processing model gets inherently more complex.
In this paper, we evaluate the security risks of MathML. We created a semi-automatic test suite and studied the JavaScript code execution and the XML processing in MathML. We added also the Content-Type handling of major browsers to the picture. We discovered a novel way to manipulate the browser's status line without JavaScript and found two novel ways to execute JavaScript code, which allowed us to bypass several sanitizers. The fact, that JavaScript code embedded in MathML can access session cookies worsens matters even more.

Keywords: MathML; Web Security; XSS

1 Introduction

HTML has largely contributed to the success of the World Wide Web. HTML can be enriched with custom styles and scripts. With the current release of HTML5 even more technologies are meant to be processed by web browsers natively. SVGs [Da11] are one example where the integration into the web browser resulted in novel security threats. These have received considerable attention in the meantime from the literature [Za12], the academic field [He11b] and the community [He11a, DW17].
In this paper, we discuss MathML [Au14] which is also natively integrated into HTML5, however, has not received any attention from the research community yet. Developers and users are waged with uncertainty about the question which risks must be dealt with due to the support of MathML, both in popular hosting sites such as Wikimedia [wi15] and in sanitizers [Ah16]. We contribute a systematic and thorough investigation of the MathML specification and implementation in popular browsers and answer the following research questions: **RQ1:** Which elements and attributes of MathML can be considered dangerous? **RQ2:** How does the handling of Content-Types by major browsers affect the security of MathML? **RQ3:** How do sanitizers and websites deal with MathML?

Officially, Firefox and Safari implement MathML [Mo17b]. However, our evaluation demonstrates that both Internet Explorer and Chrome have already implemented the

[1] Ruhr-University Bochum, christopher.spaeth@rub.de

doi:10.18420/sicherheit2018_09

processing (parsing) of MathML. According to a publicly available report there is also a prototype implementation for Chrome available [Wa17]. Currently, Google Blogger [2], WolframAlpha[3], fmath.info[4], mathjax[5] and several other Math-related websites (e.g. mathmlcentral.com[6]) support MathML.
We investigated which of the elements and attributes can be misused for script execution (and thereby Cross-Site Scripting (XSS) attacks). We identified two novel XSS variants based on MathML.
Furthermore, we uncovered a flaw in the MathML specification leading to a Status Line Manipulation attack. This means, every browser which implements the specification is susceptible to this attack. This can be used to trick the user into executing unwanted actions (e.g., redirecting to an attacker's site). Currently, for such an attack to work, JavaScript code execution is necessary (e.g. by using an event handler which overwrites the default triggered action on the fly). Our approach improves on existing attacks in respect to that no script execution is necessary.
We investigated the handling of Content-Types by major browsers. This is important as MathML is both embeddable in HTML and XML. So the question arises, how is a document handled which has a MathML file extension (*.mml*), a XML MIME-Type (*application/xml*) and *HTML* markup as content? By using a semi-automatic test suite we provide a detailed insight into the handling of documents by browsers and investigate the XSLT processing. Additionally, by extending the scope to *Namespaces*, we show how MathML can be used to execute scripts and thereby access to authentication tokens, such as cookies, is possible.
To measure the impact of our vectors, we applied them to Web Application Firewalls and sanitizers. Our investigation yields two results: Firstly, currently none of the evaluated sanitizers implement specific actions to handle MathML but rather rely on general detection mechanisms. Secondly, when benign MathML is allowed in these sanitizers, 50 % can be bypassed using at least one of our vectors. No sanitizer detects the Status Line Manipulation attack. The evaluation of a sample of websites, which currently use MathML, supports our claim that the relation between MathML and security implications is not obvious to developers yet. We found one reflected and one DOM-based XSS vulnerability.
In summary, we deliver the following contributions.
• We developed a semi-automatic test tool to investigate the security of MathML regarding Script Execution, XML (DTD) and XSLT processing.
• We found two novel XSS vectors and a scriptless Status Line Manipulation attack.
• We extend our analysis to Content-Types and demonstrate that XSLT can be processed within MathML and cookie access is possible.
• Our evaluation shows that our identified attacks can be used to bypass state-of-the-art sanitizers.

[2] https://www.blogger.com
[3] http://www.wolframalpha.com/
[4] http://fmath.info/
[5] https://www.mathjax.org/
[6] http://www.mathmlcentral.com/

2 Technical Background

2.1 MathML

MathML is a markup language to express mathematical statements. MathML can be embedded within XML (XHTML) or HTML. Therefore, it is processed by an XML or HTML parser which transforms the input ”tag soup” to a structured output. Also HTML and XHTML - as well as other languages - can be embedded into MathML. Currently, MathML is officially supported by Firefox and Safari [ca17].

The Presentation Markup is used to construct mathematical expressions which can be rendered and displayed on screen. Token elements represent the smallest unit of an expression and are, for example, combined in layout elements, such as a fraction. Consider for example List. 1 which shows how a fraction would be constructed in MathML.

```
1 <math>
2 <mfrac>
3 <mn>2</mn>
4 <mn>3</mn>
5 </mfrac>
6 </math>
```

List. 1: Example of Presentation Markup

```
1 <math>
2 <apply>
3 <csymbol>times</csymbol>
4 <ci>a</ci>
5 <ci>b</ci>
6 </apply>
```

List. 2: Example of Content Markup

The element *mfrac* starts a fraction and expects exactly two arguments - a numerator and a denominator. The rendering of this markup would be similar to $\frac{2}{3}$. Of course there are a lot more expressions available, such as root ($\sqrt{2}$), sub- (i_1) and superscript (2^3).

The Content Markup is used to describe mathematical semantics unambiguously. Consider for example the multiplication of two operands. These could be represented as *a x b*, *a*b* or simply *ab*. A human reader familiar with common mathematical notations can probably infer the meaning from the context. However, a machine can not do so. Therefore, Content Markup offers a way to represent the semantics of a mathematical expression. The example in List. 2 demonstrates how to express a product in MathML.

The element *apply* expects as its first parameter an operator which is the multiplication-operator in this case. Depending on the chosen operator one or two additional parameters are expected - in this case at least two coefficients are needed.

MathML attributes can be of 18 different types, such as string, number, color or URI [Au14]. These can be applied to either Presentation, Content Markup or both, depending on the attribute. They influence, for example, the display of MathML (such as spacing, font), reference additional semantic resources for the elements or link ressources.

2.2 Cross-Site Scripting

Web-Applications are written to interact with the user. This interaction on the client-side is mainly achieved by the use of JavaScript, which can be used to read and write values of elements and properties. The JavaScript is scoped to an origin, which is a tuple of protocol, domain and port. Cross-Site Scripting (XSS) is a code injection attack, where an attacker can make the Web-Application execute his code. Thereby, this can lead to the unwanted modification of the website or theft of authentication tokens (e.g. cookies).

3 Methodology

We executed the tests for Firefox (55.0.3), Chrome (61.0), Internet Explorer (11) on Windows 7. The tests for Safari (10.1.1) were executed on a MacBook Pro Retina with OS X (10.12.6). On the server-side an Ubuntu 16.04 with Apache 2.4.18 was used in the default configuration.

3.1 Threat Model

We refer to the Web Attacker model [Ak10] and assume that the attacker can interact with any web application on the Internet (e.g. upload files, create posts) or host its own website. He cannot intercept or inject traffic into the victim's network connection. Web Applications may be protected by Web Application Firewalls or Sanitizers. The victim will freely interact with these web applications. An attack is considered successful if an attacker can bypass the security measures and execute scripts in the originating domain.

3.2 Script Execution

According to Section 6.4.3 [Au14], MathML can be parsed by either an HTML or an XML parser. Within certain MathML elements, such as *mtext*, *mo*, *mn*, *mi*, *ms*, *annotation* or *annotation-xml*, HTML elements are allowed and processed. We verified these specification guidelines in all browsers and particularly checked for script execution, XSLT and XML processing.

We investigated all 41 elements of the Presentation and eleven basic elements from the Content Markup. Additionally, we included the elements *semantics*, *annotation* and *annotation-xml*, which belong to both the Presentation and Content markup. We supplied JavaScript code (called %vector) as content of a MathML element as shown in List. 3.

The selection process for attributes was way more complex because of the exhaustive number of attributes and attribute data types available. Some attributes are available for all elements (both Presentation and Content Markup), some are only available for either one of those and

some may only be available for selected elements. Therefore, we considered it appropriate to first analyze the attribute data types. Regarding script execution we primarily focused on the ones of type URI. This decision is based on the fact that this type is analogous to the URI type of HTML elements (e.g. <a href=...). We verified that *href* (and *xlink:href* respectively) can be used for script execution. Other attributes of type URI (math: (altimg | cdgroup | macros); mglyph: src; annotation: (definitionURL | src)) are currently not implemened by any of the browsers. To test URI attributes we supplied the vector *javascript:alert(1)* into the attribute value, as shown in List. 4. Then we clicked on the link. If an alert window opens we consider the attribute to have scripting capabilities. Plese refer to Table 1 for the list of elements and attributes which were found to be susceptible to script execution. The complete listing can be found in the extended version[7].

The element *maction* can bind an action to an expression. It has an attribute *actiontype* with the values: toggle, statusline, tooltip and input. We checked all of these values and will elaborate on the results in section 7. The elements *semantics*, *annotation* and *annotation-xml* facilitate the insertion of supplementary information for a mathematical expression. Each of these elements can declare the *encoding* of its content by using an eponymous attribute. Since MathML can be embedded in HTML and XML contexts, we focused our investigation on related Content-Types. We assigned the attribute *encoding* the values "text/html", "application/xml" and "application/xhtml+xml". To cross-reference the implications with the chosen Content-Type of the containing document, we embedded each attribute value combination into an .html, .xml and .xhtml file to observe differences. This is shown in List. 7 (c.f. Appendix).

```
1 <math>
2 <mi>2 %vector </mi>
3 <math>
```

List. 3: Test Methodology for Elements

```
1 <math>
2 <mi href="%vector">2</mi>
3 <math>
```

List. 4: Test Methodology for Attributes

3.3 Content-Types

Although several browser vendors provide online documentation about how the browser treats certain MIME types [Mo08, Mi, We] these resources do not reflect important details. For our evaluation we take the file extension, HTTP Content-Type Header and selected elements and namespaces into account. In more detail, we check the file extensions *.html*, *.xml*, *.xhtml* and *.mml*. We evaluate if the file extension or the Content-Type header (*application/xml*, *application/xml+xhtml*, *text/html*) has precedence. To make sure the results are not influenced by the content of the file, we create one version with a random element (e.g. <greeting>) and one with an HTML element (<b>). Additionally, we observe the

[7] https://goo.gl/vqyY2i

impact of the XHTML namespace on XML-based Content-Types. Our results are described in section 4.

Limitations. We leave the investigation of mobile browsers with MathML support, such as UC Browser for Android, iOS Safari, Blackberry and Opera Mini as future work. We focused on Content-Types of technologies closely related to MathML and did not consider the remaining majority of available Content-Types, since we believe that this is a research paper on its own.

4 Content-Type Handling

As elaborated in section 2 MathML can either be processed by an HTML or XML parser. In order to understand MathML's processing, it is important to first understand the general processing heuristics of HTML and XML in web browsers. Furthermore, we then apply our results to the processing of MathML. We consider the file extension (.html, .xml, .mml), the Content-Type (text/html, application/xml, application/xml+xhtml, text/mathml) and the MathML namespace (http://www.w3.org/1998/Math/MathML)

Our results show that the HTTP Content-Type header always takes precedence over the file extension. Generally speaking, if the Content-Type is set to *text/html* the browser processes the contents in an HTML context. The standard file extension for MathML *.mml* is associated with Content-Type *text/mathml*. It is quite surprising that none of the browsers render this Content-Type but rather offer to download the document with the Download Manager. We will now elaborate on the different browser behaviors when the Content-Type is either unknown or *application/xml(+xhtml)*.
When the Content-Type is unknown (no HTTP Header, file extension, known elements) Firefox displays the content as tree-view, Chrome/Safari output the content of the file as plaintext within *pre* tags, Internet Explorer interprets the content as HTML (not placed in pre tags).
By default, all browsers display a document with Content-Type *application/xml* as a tree-view (XML-context). Also, it is common knowledge that the Content-Type *application/xml+xhtml* is associated with XHTML [We] and therefore facilitates script execution.
When a not well-formed XML document is delivered, Firefox will raise an error and abort the processing, Chrome/Safari will raise an error but output the content of the document until the first error occurs, Internet Explorer outputs the content of the element as text.
We found out that all browsers upgrade an XML document to an HTML context, if an XHTML namespace is added. This facilitates the execution of JavaScript code. In Chrome, Safari and Internet Explorer this is even more problematic as this allows access to properties such as document.cookie. This way an attacker could steal authentication tokens from the victim.

Attack Scenario: Script Execution in XML. For illustration purposes, consider a web-application which accepts *.xml* files for upload. If a user accesses this file, it is delivered with Content-Type *application/xml* and displayed as tree view. Assuming an attacker includes an XHTML namespace, this can lead to XSS and cookie theft.

5 XSLT Processing

We investigated if an XML and/or HTML parser process XSLT. We found that XSLT processing is only possible, when a document is processed by an XML Parser. In our study, this is fulfilled when the Content-Type is set to *application/xml* or *application/xhtml+xml*. XSLT execution is not possible in documents which are delivered as Content-Type *text/html*. Our studies show that a downloaded *.mml* file which is subsequently opened in Firefox, also has XSLT processing capabilities.
Furthermore it is interesting to consider how an XSLT interacts with the Same-Origin Policy. Our tests show that all major browsers allow the reference of an same-origin XSLT stylesheet. None of the browsers, however, allows the inclusion of an XSLT from a foreign origin. Firefox is the only browser to process an inline XSLT styelesheet. In the past [He11b] this could be used to create an SVG Chamaeleon. A similar attack is possible with MathML, as shown in List. 8.

6 Script Execution in MathML

We checked which elements of MathML support scripting capabilities and should therefore be considered potentially dangerous. The complete results are listed in the extended version.

All Browsers. Our investigation shows that the elements *mn*, *mi*, *mo*, *ms* and *mtext* (not in IE) have scripting capabilities. It should be noted that this does not apply if the script element is a child of the parent *math* element. This insight applies to all browsers - even those which do not officially implement MathML. We can conclude from this fact, that also browsers, such as Chrome and Internet Explorer, already implement the processing of MathML as part of HTML5. It should also be taken into account that the parsing context switches to HTML if an HTML element is found outside the previously mentioned MathML elements. Hence, all further MathML elements are no more in the MathML scope. Therefore, this could also be used to trigger script execution.

Our investigation shows that the *href* and *xlink:href* [He11b] attribute is susceptible for script execution - for example by using the well-known *javascript:* pseudo-protocol. Additionally, we adapted a vector from [Ma17] using an *xml:base* and *href* attribute for the use with MathML. The vectors are provided in List. 5.

```
1 <math href="javascript:alert(1)"> <mi>2</mi></math>
2 <math><mi xml:base="javascript:alert(1)//" href="#">2</mi></math>
```

List. 5: An Vector Based on xml:base which can be used to test for XSS

Script Execution in *annotation*, *annotation-xml* and *semantics* depends on the value of the *encoding* attribute and on the Content-Type of the document. Our evaluation shows that 1. script execution is largely not possible within an HTML document. There are two exceptions: When the attribute *encoding* has either the value *text/html* or *application/xhtml+xml*. 2. if the Content-Type of the host document is *application/xml* or *application/xhtml+xml* scripts are executed in all of the elements irregardless of the chosen value for the attribute *encoding*. This is quite surprising, as one would expect that JavaScript code is executed within an HTML document.

No combination of a document's Content-Type and the value of the *encoding* attribute can be used to trigger XML Entity Attacks or the processing of a XSLT, which is supplied as the child element of the elements *annotation*, *annotation-xml* and *semantics*.

7 Status Line Manipulation Attack

The *maction* element has the attribute *actiontype*. This attribute can be assigned the values: *toggle*, *tooltip*, *input* and *statusline*. The value *toogle* displays different subexpressions. *tooltip* displays a tooltip when hovering over the expression and *input* facilitiates modification of the expression. The value *statusline* modifies the browser's status line with a stored text. Before discussing any details of the attack, one should consider the implications of being able to modify the status line. The status line is used to show the target of a hyperlink (i.e. value of the attribute *href*). This provides the user with additional information, which action the browser is going to take after clicking the link. Therefore, the correct display of the value is clearly security relevant. Malicious websites may add a JavaScript event handler (onClick), which executes a different action despite displaying the correct destination of the link. This facilitates redirecting users to an arbitrary destination or executing unwanted actions. Our attack differs from existing work in that no script execution is necessary but can be achieved solely with MathML. A proof of concept code for Firefox is provided in List. 6.

```
1 <html> <body>
2 <math href="http://attacker.com/target.html">
3 <maction actiontype="statusline">
4 <mfrac><mn>1</mn><mn>2</mn></mfrac>
5 <mtext>http://www.w3.org/TR/MathML3/chapter3.html#presm.mfrac</mtext>
6 </maction>
7 </math></body></html>
```

List. 6: A Scriptless Status Line Manipulation Attack with the Element maction

While the browser will display the value of the element *mtext*, implying a reasonable resource as the target of the link, when clicking the link the user is redirected to attacker.com. Execution of arbitrary JavaScript is also possible. by using the *javascript:* pseudo protocol. This issue has been reported to Mozilla [Ch17].

8 Evaluation of Sanitizer and Web-Application Firewalls

To show the feasibilty of our attacks, we tested our vectors against a selected set of Web Application Firewalls and PHP-based sanitizers. In detail we considered Modsecurity CRS, RaptorWAF, HTMLPurifier and HTMLSafe. First of all, we checked if the inclusion of JavaScript inside of MathML (e.g. `<math><mi><script>...</script></mi></math>`) can be used to bypass any of the aforementioned sanitizers. Additionally, we tested the vectors of List. 5 and List. 6. We excluded the testing of XSLT because sanitizing is usually applied in an HTML context and the resulting page will be of Content-Type *text/html*, essentially making the XSLT instructions void. Our tests show that including a *script* element inside a MathML element does not yield any advantage in bypassing Sanitizers compared to supplying the vectors in plain.

HTMLSafe [Go10] is a sanitizer available for PHP. It implements a combination of black- and whitelists. According to our source code analysis and tests, HTMLSafe blacklists the *javascript* protocol and does not whitelist the *xml:base* attribute by default. Therefore both XSS vectors are blocked. However, the Status Line Manipulation attack passes.
HTMLPurifier [ht17] is one of the recommended ways for sanitizing HTML markup with PHP and is a whitelist based sanitizer. Albeit, MathML is currently not implemented in the whitelist. Therefore, by default, even benign MathML is blocked and of course our vectors. To investigate the possible implications of allowing arbitrary MathML elements, we created a prototype which whitelists the elements *math*, *mn*, *mfrac*, *mi* and *maction*. We did not do any further modifications to the source code. Additionally, we whitelisted the attribute *href* for the element *math*. Our tests show that if the attribute *href* is whitelisted as data type CDATA, HTMLPurifier does not sanitize the value and a bypass is possible. In order to do sanitization correctly, the attribute *href* has to be of type URI.
RaptorWAF [Co17] is a Web-Application Firewall. It can be easily bypassed with both vectors by supplying a HTML encoded version of the vectors. Additionally, certain keywords, such as alert and script, must not be used or send with different spelling (i.e. SCRipt). The Status Line Manipulation vector also passes.
Modsecurity with the Core Rule set [Mo17a] is a freely available Web-Application Firewall. It has to be modified[8] in order to allow benign MathML. Both XSS vectors are blocked. However, the Status Line Manipulation vector passes.

We conducted a small sample of tests with our vectors on websites which currently use MathML. We found that both the live demo on mathjax.org and MathJax Sandbox [9] are

[8] Disable rules: 950901, 981173, 900048
[9] http://jbergknoff.github.io/mathjax-sandbox/

susceptible to script execution by using vector 1 from List. 5. While the former is not exploitable, the latter has a DOM-XSS vulnerability. MathMLCentral [10] processes an uploaded file without sanitation. It is vulnerable to reflected XSS, which can exploited with all vectors. Google Blogger[11] supports the usage of MathML markup. While both XSS vectors would only lead to "Self-XSS" a malicious blogger could use the Status Line Manipulation attack to redirect users to unwanted locations.

9 Related Work

To our knowledge there is no academic publication available dealing with the security of MathML. Heiderich et al. [He11b] have investigated the dangers of SVGs which motivated this work and some attacks could be directly applied. Heiderich has also reported on scriptless attacks by abusing Cascading Style Sheets [He12] Barth et al. [BCS09] have reported on the dangers of MIME Sniffing in browsers. However, they have neither investigated the precedence of file extensions, Content-Types and Namespaces. Various posts on the Internet appeared in the past, discussing MathML-based XSS vectors [Pa17, ja17, Sp17, cu]. DOMPurify [HSS17] - a DOM-based Sanitizer - implements the sanitization of HTML, SVG and MathML. Although these resources contribute to the public awareness of MathML and its potential dangers, they do not provide a systematic and thorough investigation of MathML.

10 Conclusion

Regarding our question if *MathML is dangerous* we can conclude with the following facts to consider: MathML as a standard embedded in HTML5 is implemented by all major browsers. Therefore any security issue found will affect a large user base. Due to the embedding of MathML in a HTML context all elements can be used for script execution. Furthermore XSLT execution might constitute a security issue. The scriptless manipulation of the status line should also be taken into account. MathML is a novel threat, which should be taken seriously. We encourage the community to extend our work and investigate the support of MathML in other software, such as accessibility software and editors.

11 Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research was supported by the German Ministry of research and Education (BMBF) as part of the SyncEnc research project.

[10] http://www.mathmlcentral.com/Tools/FromMathMLFile.jsp
[11] https://www.blogger.com/

References

[Ah16] Ahmady, Abdul: , Implementation of MathML DTD3. http://htmlpurifier.org/phorum/read.php?5,8091, 2016.

[Ak10] Akhawe, Devdatta; Barth, Adam; Lam, Peifung E; Mitchell, John; Song, Dawn: Towards a formal foundation of web security. In: Computer Security Foundations Symposium (CSF), 2010 23rd IEEE. IEEE, pp. 290–304, 2010.

[Au14] Ausbrooks, Ron: , Mathematical Markup Language (MathML) Version 3.0 2nd Edition. https://www.w3.org/TR/2014/REC-MathML3-20140410/, 2014.

[BCS09] Barth, Adam; Caballero, Juan; Song, Dawn: Secure content sniffing for web browsers, or how to stop papers from reviewing themselves. In: Security and Privacy, 2009 30th IEEE Symposium on. IEEE, pp. 360–371, 2009.

[ca17] caniuse: , MathML. http://caniuse.com/mathml, 2017.

[Ch17] Christopher Späth: , MathML maction statusline - status bar text doesn't accurately reflect the target of the link. https://bugzilla.mozilla.org/show_bug.cgi?id=1392258, 2017.

[Co17] CoolerVoid: , Raptor - WAF - Web application firewall using DFA. https://github.com/CoolerVoid/raptor_waf, 2017.

[cu] cure53: , HTML5 Security Cheatsheet. https://html5sec.org/.

[Da11] Dahlström, Erik: , Scalable Vector Graphics (SVG) 1.1 (Second Edition). https://www.w3.org/TR/2011/REC-SVG11-20110816/, 2011.

[DW17] DW: , What does a HTML filter need to do, to protect against SVG attacks?, 2017. https://security.stackexchange.com/questions/26264/what-does-a-html-filter-need-to-do-to-protect-against-svg-attacks/30390.

[Go10] Gocobachi, Miguel: , Package Information: HTML_Safe. https://pear.php.net/package/HTML_Safe, 2010.

[He11a] Heiderich, Mario: , The image that called me, 2011. https://www.owasp.org/images/0/03/Mario_Heiderich_OWASP_Sweden_The_image_that_called_me.pdf.

[He11b] Heiderich, Mario; Frosch, Tilman; Jensen, Meiko; Holz, Thorsten: Crouching tiger-hidden payload: security risks of scalable vectors graphics. In: Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp. 239–250, 2011.

[He12] Heiderich, Mario; Niemietz, Marcus; Schuster, Felix; Holz, Thorsten; Schwenk, Jörg: Scriptless attacks: stealing the pie without touching the sill. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, pp. 760–771, 2012.

[HSS17] Heiderich, Mario; Späth, Christopher; Schwenk, Jörg: DOMPurify: Client-Side Protection Against XSS and Markup Injection. In: European Symposium on Research in Computer Security. Springer, pp. 116–134, 2017.

[ht17] htmlpurifier: , htmlpurifier. http://htmlpurifier.org, 2017.

[ja17] jackmasa: , Math. https://twitter.com/jackmasa/status/930096423168655361, 2017.

[Ma17] Masato Kinugawa: , SVG xml base. https://twitter.com/kinugawamasato/status/898950198826721280, 2017.

[Mi] Microsoft: , MIME-Handling Changes in Internet Explorer. Accessed: 31.01.2017.

[Mo08] Mozilla: , How Mozilla determines MIME Types. https://developer.mozilla.org/en-US/docs/Mozilla/How_Mozilla_determines_MIME_Types, 2008. [Online; accessed 31-January-2018].

[Mo17a] Modsecurity: , ModSecurity. http://modsecurity.org/, 2017.

[Mo17b] Mozilla: , MathML. https://developer.mozilla.org/en-US/docs/Web/MathML, 2017.

[Pa17] Payloads, XSS: , Filter evasion. https://twitter.com/XssPayloads/status/935051449670750209, 2017.

[Sp17] Späth, Christopher: , MathML xml base. https://secanalysis.wordpress.com/2017/08/28/mathml-xmlbase/, 2017.

[Wa17] Wang, Frederic: , Review of Igalia's Web Platform activities, 2017. http://frederic-wang.fr/review-of-igalia-s-web-platform-activities-H1-2017.html.

[We] Webkit: , Understanding HTML, XML and XHTML. Accessed: 31.01.2017.

[wi15] wikimedia: , Consider opening up POST /media/math/format to external users. https://phabricator.wikimedia.org/T116147, 2015.

[Za12] Zalewski, Michal: The tangled Web: A guide to securing modern web applications. No Starch Press, 2012.

A Results

A.1 Semantics, Annotation and Annotation-xml

```
1  <math xmlns="http://www.w3.org/1998/Math/MathML">
2  <semantics encoding="text/html">
3  <iframe xmlns='http://www.w3.org/1999/xhtml' src='javascript:alert(1);'></iframe>
4  </semantics>
5  <semantics encoding="application/xml">
6  <iframe xmlns='http://www.w3.org/1999/xhtml' src='javascript:alert(1);'></iframe>
7  </semantics>
8  <semantics encoding="application/xhtml+xml">
9  <iframe xmlns='http://www.w3.org/1999/xhtml' src='javascript:alert(1);'></iframe>
10 </semantics>
11 </math>
```

List. 7: Script Execution Test for element semantics; tests for elements annotation and annotation-xml are constructed analogous

A.2 MathML Chamaeleon

```
1  <?xml−stylesheet type="text/xml" href="#style1"?>
2  <math>
3  <mfrac><mi>2</mi><mi>3</mi></mfrac>
4  <xsl:stylesheet id="style1" version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
       Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format">
5  <xsl:template match="/">
6  <html xmlns="http://www.w3.org/1999/xhtml">
7  <b>testing</b>
8  </html>
9  </xsl:template>
10 <xsl:template match="xsl:stylesheet"></xsl:template>
11 </xsl:stylesheet>
12 </math>
```

List. 8: A MathML Chamaeleon

A.3 Dangerous Elements and Attributes

For all tables: A ”0” means "no script execution, while ”1” means ”script execution”.

Tab. 1: Only elements are listed in which at least one element in a browser is susceptible to script execution. Attributes are tested with attribute href;

Element	Firefox	IE 11	Chrome	Safari
mi	1	1	1	1
mn	1	1	1	1
mo	1	1	1	1
mtext	1	0	1	1
ms	1	1	1	1
Attributes	1	1	1	1
cn	1	0	0	1
ci	1	0	0	1
csymbol	1	0	0	1
cs	1	0	0	1
apply	1	0	0	1
bind	1	0	0	1
bvar	1	0	0	1
share	0	0	0	0
semantics	1	0	0	1
cerror	1	0	0	1
cbytes	1	0	0	1

Attributes	FF	IE 11	CH	SA
mglyph	1	0	0	0
mi	1	0	0	1
mn	1	0	0	1
mo	1	0	0	1
mtext	1	0	0	1
ms	1	0	0	1
mrow	1	0	0	1
mfrac	1	0	0	1
msqrt	1	0	0	1
mroot	1	0	0	1
mstyle	1	0	0	1
merror	1	0	0	1
mpadded	1	0	0	1
mfenced	1	0	0	1
msub	1	0	0	1
msup	1	0	0	1
msubsup	1	0	0	1
munder	1	0	0	1
mover	1	0	0	1
munderover	1	0	0	1
mmultiscripts	1	0	0	1
mprescripts*	0	0	0	1
none*	0	0	0	1
mtable	1	0	0	1
mtr	1	0	0	1
mlabeledtr	1	0	0	1
mtd	1	0	0	1
maligngroup	0	0	0	1
malignmark	0	0	0	1
mstack	1	0	0	1
mlongdiv	1	0	0	1
msgroup	1	0	0	1
msrow	1	0	0	1
mscarries	1	0	0	1
mscarry	1	0	0	1
msline	1	0	0	1
math	1	0	0	1
maction	1	0	0	0
semantics	1	0	0	1
annotation	1	0	0	0
annotation-xml	1	0	0	0

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 133

Harmonizing physical and IT security levels for critical infrastructures

Vanessa Chille[1], Sibylle Mund[2], Andreas Möller[3]

Abstract: We present a concept for finding an appropriate combination of physical security and IT security measures such that a comprehensive protection is provided. In particular, we consider security for critical infrastructures, such as railway systems. For classifying physical security measures, the so-called Protection Classes from the standard EN 50600 are used in our approach. To provide comprehensive protection for a system under consideration, these sets of explicit physical security measures need to be combined with other kinds of security, such as IT security and organizational security. We present a new classification approach named 'Type of Attack(er)' that allows for taking all aspects of security into joint consideration, and harmonizes physical and IT security levels by creating a link between EN 50600 and IEC 62443.

Keywords: physical security, IT security, IEC 62443, EN 50600, critical infrastructures

1 Introduction

In recent times, threats for critical infrastructures have attracted increasing attention. The motivation and character of suspected attacks differ greatly. They range from vandalism through cyber attacks that are not specifically targeted towards a particular organization to international terrorism. A typical example for a critical infrastructure - requiring security measures to ensure not only the system's availability but also the safety of people – is a railway system.

A comprehensive approach towards security necessitates the consideration of a number of different aspects of security. Beside the aspect of IT security that is being treated with increasing care by now, physical security plays an important role. It represents a necessary complement to enable overall security that deserves more attention than it often receives. Appropriate physical security measures prevent two different types of attacks: purely physical attacks and attacks on the IT system enabled by physical access (for example to a USB port). In the latter case, physical security measures represent an important additional security perimeter complementing IT security measures. Purely physical attacks may consist of damaging equipment or performing other manipulations, mostly causing an impairment of the system's availability and financial losses. Even though the consequences may be less severe for the latter case, the operator of a critical infrastructure will have great interest in avoiding the effects of both types of attacks.

[1] Siemens AG, Mobility Division, Ackerstraße 22, 38126 Braunschweig, vanessa.chille@siemens.com
[2] Siemens AG, Mobility Division, Ackerstraße 22, 38126 Braunschweig, sibylle.mund@siemens.com
[3] Siemens AG, Mobility Division, Ackerstraße 22, 38126 Braunschweig, andreasmoeller@siemens.com

doi:10.18420/sicherheit2018_10

For this purpose, appropriate measures have to be identified, which requires an evaluation of the effectiveness of the measures (Sec. 2). Furthermore, the security of a system depends strongly on how well all aspects of security are coordinated and work together. Therefore, only a holistic point of view can lead to comprehensive protection (Sec. 3). In the end of the present paper, we also comment on how such protection can be achieved for the example of railway systems (Sec. 4). The first step in this direction is to find a possibility for classifying physical security measures. Only few works exist that address the topic of physical security in a detailed and comprehensive way. Standards for particular components such as doors and windows [EN27], cylinder locks [EN03] and the like go in the very details, and for instance even comment on testing procedures. Standards addressing security on a global level make mostly only general statements about how to achieve physical security and do not give explicit requirements or precise measures that should be taken. For example [NE-4d] gives a number of organizational measures that shall be taken, but does not go into detail about requirements for measures intending to prevent unauthorized access. Another prominent example is ISO/IEC 27001 [ISO01], which, for instance, states that physical security perimeters shall exist, but does not elaborate on how to implement them. In IEC 62443 [IEC3-3], the physical security measures suggested as compensating countermeasures are not specified either. In that context, it is also necessary to understand which physical security measures correspond to which IT security measures. Furthermore, there are the German publications VdS 2007 [VdS07] and VdS 2333 [VdS33], the combination of which provides a consistent and detailed concept for sets of physical security measures. International standards are however to be preferred in an international context. A vast amount of general literature exists that comments on how to implement measures such as video surveillance or physical barriers in an appropriate way for particular sites. Most of it, however, does not provide any real classification of physical security measures but rather comments on principles. A very convenient standard in that context is EN 50600 ([EN-1] and [EN2-5]): we found that it is also well-suited for our purposes, and describe its main aspects in Sec. 2. We use the physical security classification from this standard as a tool in our approach. To the best of our knowledge, a systematic and holistic approach to physical security, in particular for critical infrastructures, has been missing so far. We present an approach that is suitable for critical infrastructures, and also for the very special conditions of railway systems. Our concept uses existing standards and, in particular, unites IEC 62443 and EN 50600. For that purpose, we introduce a new classification named ‘Type of Attack(er)’ that captures all aspects of security at once (see Sec. 3.2). It might be regarded as a continued development of the holistic security concept discussed in [Ko16] by adding the aspect of physical security.

An illustration of this idea can be found in Fig. 1, where the most important kinds of security and their interplay are depicted. It is indicated that the basis of all security is the process maturity that describes an organization’s capability to follow procedures. All other security measures are in vain if one cannot rely on the organization to implement the required organizational security measures. The Maturity Model from IEC 62443-2-4 [IEC2-4] can be utilized for the evaluation. It uses four Maturity Levels; from Maturity Level 3 on, the performance is repeatable, i.e. the organization is able to actually adhere

to processes. IT security and physical security are built on the foundation formed by process maturity and need to complement each other. Both of them also contain organizational security measures. IT security is addressed in IEC 62443 and classified by Security Levels (SL). The topic of physical security shall be addressed in detail in the present paper. The joint effect of all these aspects of security shall be the resistance against particular attacks or attackers, classified by the 'Type of Attack(er)'.

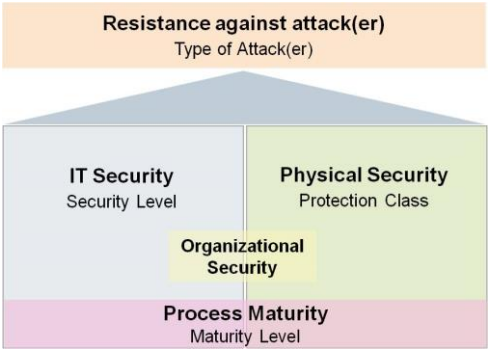


Fig. 1: Different aspects of security and their interplay

2 The standard EN 50600

The standard EN 50600 is not the obvious standard to be used for critical infrastructures, as it provides regulations for data centre facilities and infrastructures. However, an examination of the details reveals that it is well-suited for ensuring the physical security in any kind of building.

2.1 Basic principles

EN 50600 gives a practical concept for the implementation of physical security using common security principles [EN-1]. Firstly, a risk assessment is to be performed that is then used as the foundation of decisions on which security measures shall be taken. For this reason, it is frequently referred to throughout the standard. Furthermore, EN 50600 also explicitly addresses the topic of organizational security and requests organizational measures to accompany physical security. Another concept appearing also in many other contexts related to security is Defense in Depth. It means that multiple security measures shall be taken for the protection of the assets such that an attacker cannot intrude by overcoming one single security measure. For physical security, it means quite literally that security perimeters consisting of physical barriers shall be arranged in an onion skin-like configuration (see Fig. 2). EN 50600 utilizes a system of four Protection Classes (PC). The assets that require the strongest protection shall be located in PC 4, i.e. the highest class, where the criteria for gaining access are the most restrictive.

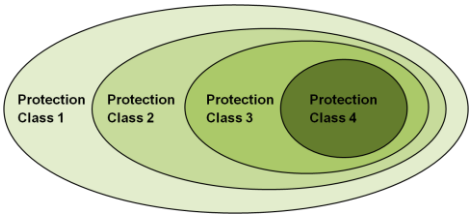


Fig. 2: Concept for the Defense in Depth principle on the level of physical security [EN-1]

2.2 Protection of boundaries

The boundaries of the areas associated with the various Protection Classes (as indicated in Fig. 2) are supposed to be protected by means of passive elements, i.e. mechanical barriers, as well as technical security systems for the prevention of unauthorized access. The latter are further described by referring to specialized standards for the respective systems: security lighting, video surveillance systems (EN 62676-1-1:2014, Grade 2, where justified according to the risk assessment), intruder and holdup alarm systems (EN 50131, security grade according to risk assessment), access control systems (EN 60839-11-1, security grade according to risk assessment), and alarm monitoring (EN 50136 series and EN 50518 series).

The passive elements are characterized by referring to the so-called Resistance Classes (RC) from EN 1627 on burglar resistance [EN27]. Elements such as doors, windows, locks, and the like are supposed to resist the attack of a particular kind of burglar with a defined tool set and a limited amount of time. The Resistance Classes utilized in EN 50600 are illustrated on the right-hand side of Fig. 3. The tool sets get more elaborate the higher the Resistance Class; the pictures only mean to give an idea, please find the detailed lists of allowed tools in EN 1630 [EN30].

2.3 Protection Classes

The Protection Classes require different sets of the aforementioned protective measures [EN2-5]. Here, we only want to give a basic impression of the most important requirements, detailed information can be found in EN 50600 itself [EN2-5]. For the passive elements, the requirements are given in a precise way by attributing particular Resistance Classes to the Protection Classes (PC 1 – RC 2 / PC 2 – RC 3 / PC 3 & 4 – RC 4). Concerning the technical systems, the standard does not make explicit statements on their deployment in the various Protection Classes. It is sometimes however implied when, for example, from PC 2 on, the opening of an emergency door must cause an alarm by the intrusion alarm system. Another topic worth mentioning is the one of the co-location of boundaries. Areas designated to different Protection Classes need to be separated by identifiable physical barriers. Not all boundaries of the areas attributed to the various Protection Classes are allowed to be co-located, which ensures the presence

of multiple physical security perimeters around the most critical assets. One can regard this as a contribution to the fulfilment of the Defense in Depth principle.

Please note that EN 50600 offers flexibility concerning the conditions to be provided for the Protection Classes. On the basis of the risk assessment, one may decide to apply particular protective measures or not. The conditions that one is supposed to establish for the Protection Classes originate from their definitions that describe how many and which kind of people shall have access to the respective areas. They can be found in Tab. 1 below.

Protection Class 1	**Protection Class 2**	**Protection Class 3**	**Protection Class 4**
public or semi-public	accessible to all authorized personnel, employees as well as visitors	accessible only for specified employees and visitors	accessible only for specified employees with an identified need for the access

Tab. 1: Definitions of the Protection Classes via access authorizations [EN-1]

3 Comprehensive protection

3.1 Physical security & IT security

The above approach using access authorizations as the defining characteristic of the classification can pose difficulties. A system's need for protection might not always be perfectly in line with the intended limitations of access authorizations. Therefore, the approach is not the most convenient basis for the decision which of the Protection Classes is suitable for an individual system under consideration. Furthermore, the definition of the Protection Classes is an inconvenience in another way: we aim at a holistic view on security and for this purpose want to analyze the interplay between IT security and physical security. IT security is commonly classified via the Security Levels from IEC 62443 [IEC3-3]. This standard has originally been designed for industrial automation and control systems, but is being utilized in other domains as well. In order to understand the correspondence between IT security and physical security, we aim at finding correspondences between Security Levels and Protection Classes. The Security Levels are defined via a characterization of the expected attacker and his means, resources, skills and motivation. The definition of the Security Levels thus follows a philosophy that is entirely different from the aforementioned approach for the Protection Classes. However, as discussed above, for passive elements, the Protection Classes refer to the so-called Resistance Classes from EN 1627. These are again defined via a characterization of the burglar that is to be expected. The close resemblance to the definition of the Security Levels offers a convenient way of matching Protection Classes and Security Levels. One may argue about whether it is fair to base the argument for the correspondences only on the characterization of the passive elements. However, since the respective Resistance Classes are regarded as an adequate component of the

protection required for a particular Protection Class, this seems to be a legitimate approach.

Fig. 3 illustrates the matching of Security Levels and Protection Classes. PC 1 does not fit to SL 1, as the tool set available to the attacker for PC 1 is too large for a casual or coincidental attack. The low level of risk the attacker is willing to take in PC 1, however, fits nicely to the attacker's low motivation in SL 2. After a comparison of further aspects, PC 1 and SL 2 can be associated. Similar lines of argumentation lead to matching PC 2 with SL 3 and PC 3 and 4 with SL 4.

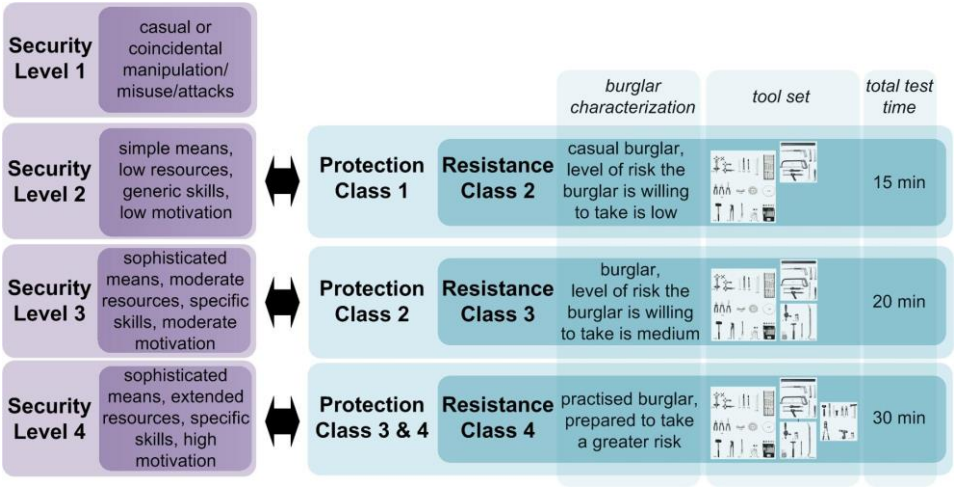


Fig. 3: Matching Security Levels [IEC3-3] and Protection Classes [EN2-5] (and Resistance Classes [EN27, EN30])

One may continue and also specify security grades for technical security systems. EN 50600 requires for most of them only that they shall be compliant with the respective specialized standards, and to choose their security grades according to the risk assessment. However, the above logic can also be pursued further and in this way security grades of technical systems are linked to the Protection Classes and Security Levels. It facilitates the mapping that in many of the associated standards, the security grades are also defined via some sort of characterization of the attacker's means, resources and the like. The results can be found in Tab. 2.

IEC 62443	EN 50600	EN 1627	EN 62676	EN 50131	EN 60839
IT security	physical security	passive elements	video surveillance	intrusion and holdup alarm	access control
SL 1		RC 1 N*	X	X	mechanical key**
SL 2	PC 1	RC 2	Grade 2	Grade 2 (or 1)	Grade 2 (or 1)
SL 3	PC 2	RC 3	Grade 3	Grade 3	Grade 3
SL 4	PC 3 or PC 4	RC 4	Grade 4	Grade 4	Grade 4

* RC 1 N protects mainly against acts of vandalism, such as attempts against forced entry by physical force (kicking, jumping, shoulder slams, lifting up and tearing out). It should thus provide enough protection to prevent an unauthorized person from casually or coincidentally accessing areas that require a minimum of protection against unauthorized manipulation.
** A conventional mechanical key seems perfectly sufficient: no sophisticated options such as unique identification of the user or multifactor authentication are needed. The management and storage of the keys need to be controlled.

Tab. 2: Correspondences between Security Levels, Protection Classes, Resistance Classes and security grades of technical systems

3.2 Type of Attacker

As we have already seen above, different aspects of security are regulated by a large number of different classifications. Most of them entail lists of detailed requirements that need to be fulfilled. This approach is very helpful when one is searching for precise guidance about how to implement the fulfillment of particular security requirements. In a first step, one however often does not want to make all these implications, as the analysis of a system's need for protection is independent of the practical implementation of protection by means of specified measures. A solution for this issue would be the introduction of a generic classification offering the possibility to express solely the level of protection. It should not make any statement on the implementation of that protection, i.e. whether, for example, means of IT, physical or organizational security are used to achieve it. In the previous section, you have seen that characterizing the attacker and his skills, motivation, tools and the like has proved to be a convenient principle allowing for finding links between different classifications. This is also due to the fact that many classifications already use it as an underlying principle. We thus suggest turning this approach into a classification itself. For this purpose, we characterize four Types of Attack(er)s (ToA) that a system can be supposed to be resistant against:

ToA 1	casual or coincidental manipulation or attack
ToA 2	attack(er) with simple means, low resources, generic skills and a low motivation
ToA 3	attack(er) with sophisticated means, moderate resources, specific skills and a moderate motivation
ToA 4	attack(er) with sophisticated means, extended resources, specific skills and a high motivation

Fig. 4: Types of Attack(er)s (ToA)

The above definition of the ToA is strongly inspired by the definitions of the Security Levels from IEC 62443 [IEC3-3]. The difference is that the ToA only expresses that a system is supposed to be resistant against that particular type of attacker. It does not imply anything more.

The approach offers a number of advantages:

- The definitions of the classification are as clear and simple as possible.
- A generic term is created that offers the possibility to talk about the level of protection required for a system without implying the deployment of particular measures.
- As many classifications already use the characterization of the attacker as an underlying principle, finding appropriate counterparts in the various domains of security is easy.
- The approach allows for taking the different aspects of security into joint consideration and is therefore truly holistic.

It is one of the key ideas of the approach that the Type of Attack(er) characterizes the attacker and does not imply anything more. By definition, there are no particular requirements associated with the ToAs. However, we offer guidance by making suggestions for how to achieve a particular ToA by linking different security classifications (see Tab. 2).

Please note that in order to provide comprehensive protection for a system, one also needs to understand how the different kinds of security work together: if they address the same factors and are thus redundant (possibly intended to ensure Defense in Depth), or if they take care of independent gateways that an attacker might exploit. We should thus aim at evaluating the joint effect of physical and IT security measures systematically. The Type of Attack(er) can be useful in that context. For illustrating the results of such an evaluation, we use a similar tool as the holistic security concept (HSC) from [Ko16]. The HSC utilizes a matrix to show which combinations of Security Levels and Maturity Levels result in which so-called Protection Levels. It expresses what kind of process maturity is mandatory to ensure that IT security measures can actually have an effect such that a particular Protection Level is achieved. The Protection Levels correspond to the Security Levels in a direct way and express that the protection targeted by the Security Levels is indeed achieved. We can develop that concept further and also take the physical security aspect into consideration by adding another axis to the matrix. A four-dimensional matrix is the result. For the sake of simplicity, for now, we want to assume that the Process Maturity is on Maturity Level 3 such that the organization is capable of following procedures. In this way, we may focus on IT security and physical security measures. Firstly, we need to ask what measures can actually be implemented to protect the system under consideration (SUC). We are facing three different categories for a *s*ystem's *c*apability of being *p*rotected (SCP) by physical or IT security measures, as illustrated in Fig. 5. Either only IT security or physical security measures, or both IT

and physical security measures can be implemented. In the case of the latter, one can furthermore distinguish the three cases (SCP3-I), (SCP3-II) and (SCP3-III) as indicated in Fig. 5. Those categories elaborate on which measures protect the same or independent ways of intrusion.

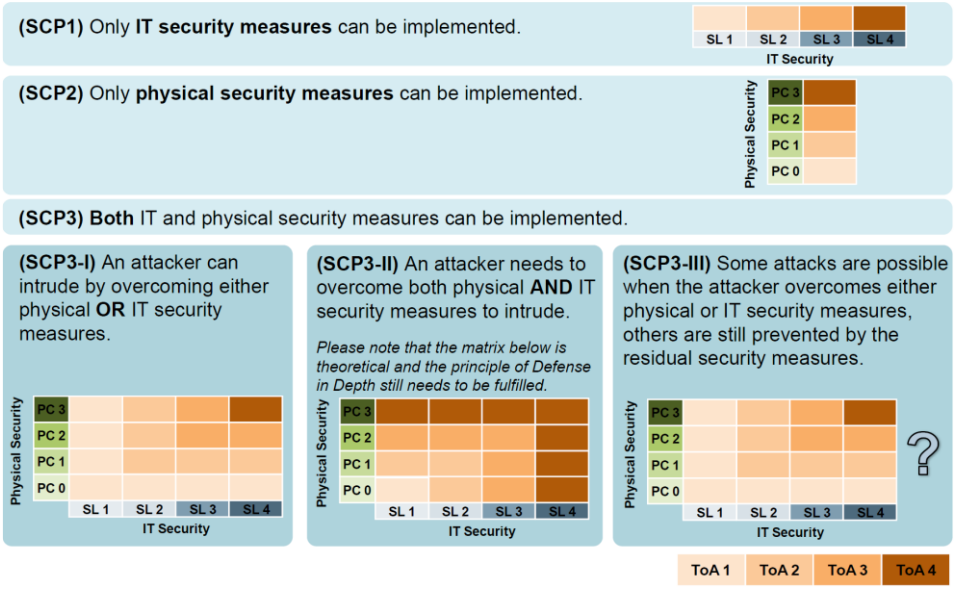


Fig. 5: Joint effect of physical and IT security measures

Fig. 5 also shows the matrices for all those cases. For (SCP3-I), both IT security and physical security measures need to be on an adequate level in order to ensure the security of the system. The measures need to be strong individually and the correspondences in Tab. 2 should be taken into consideration. This relation is also indicated in the matrix. For a system from category (SCP3-II), IT and physical security add up, such that only one of the two needs to be on a high level. Please note that the depicted matrix is purely theoretical, as the Defense in Depth principle still requires the implementation of both IT security and physical security. Systems assigned to category (SCP3-III) are the most complicated and the most common at the same time. As an example, one could think of a server room: if an attacker only overcomes the physical security measures and breaks into the building, he can perform an attack on the availability by destroying the equipment with brute force. In order to intrude into the IT system, the attacker needs to overcome further IT security measures. This situation is too complex to express it in a single matrix. In practice, most systems do in fact belong to category (SCP3-III).

Even if the above matrices thus cannot provide a comprehensive description of the complex issue in many cases, one aspect of the considerations should always be kept in mind: it is recommended to differentiate between the ‘OR’ and ‘AND’ cases (SCP3-I and SCP3-II) when designing the set of security measures for any SUC. In addition, one

should consider that in case of doubt, the Defense in Depth principle always suggests deploying all security measures according to Tab. 2.

4 Example: comprehensive protection of a railway system

In the context of railway systems, critical assets can be found in different kinds of locations thus providing very particular conditions. These may be categorized as trains, tracks and rooms - such as Operation Control Centers and server rooms. Independently of the existing differences in the locational circumstances, we need to aspire to provide the same security for the assets in all locations. This target can be expressed very easily by using the Type of Attack(er) approach: the same ToA shall be assigned to all locations. In order to decide which ToA is suitable, one may recourse to established principles, as they already exist for IT security and the assignment of Security Levels. The ToA that one assigns to a system under consideration shall match these Security Levels. It is convenient to base the decision on the ToA on the analysis for IT security, as such analyses are being performed by most organizations already (for example according to [DIN-04]). This means, for example, that if SL 3 is assigned, the system shall be protected according to ToA 3, as those classifications share the same attacker characterization (see Fig. 3 and Fig. 4). The fact that all locations shall be protected according to a particular ToA, however does not mean that the very same protective measures need to be applied everywhere. The specific conditions in the individual locations shall be taken into account to find the appropriate measures. The Protection Classes from EN 50600 are designed for the description of physical security as it can be implemented for the protection of rooms. If one intends to protect equipment in unusual areas providing special conditions, as trains or tracks, more individual solutions are required. These can be found by analyzing the individual case, consulting specialized standards as depicted in Tab. 1 and combining those mechanical housings and technical systems that are feasible and result in a protection corresponding to the assigned ToA.

5 Conclusion

Physical security measures shall always be part of a holistic security concept; this applies in particular for critical infrastructures. Depending on the individual assets and the particular conditions of their location, different explicit measures shall be taken. A useful tool for choosing the right set of security measures is the Type of Attack(er) that characterizes the attacker a system shall be resistant against. It represents a holistic approach as it allows for taking all aspects of security into joint consideration. In particular, the joint effect of IT security (IEC 62443) and physical security (EN 50600) can be analyzed in this way. In future, our approach may be used to find appropriate sets of protective measures for various specific applications. Furthermore, additional aspects of security may still be added explicitly and links to associated standards could be found.

Bibliography

[DIN-04] DIN VDE V 0831-104: Electric signalling systems for railways – Part 104: IT Security Guideline based on IEC 62443

[EN03] EN 1303: Building hardware – Cylinders for locks – Requirements and test methods, European standard, 2015

[EN-1] EN 50600-1: Information technology – Data centre facilities and infrastructures – Part 1: General concepts, European standard, 2012.

[EN2-5] EN 50600-2-5: Information technology – Data centre facilities and infrastructures – Part 2-5: Security systems, European standard, 2016.

[EN27] EN 1627: Pedestrian doorsets, windows, curtain walling, grilles and shutters – Burglar resistance – Requirements and classification, European standard, 2011.

[EN30] EN 1630: Pedestrian doorsets, windows, curtain walling, grilles and shutters – Burglar resistance – Test method for the determination of resistance to manual burglary attempts, European standard, 2016.

[IEC2-4] IEC 62443-2-4: Security for industrial automation and control systems – Part 2-4: Security program requirements for IACS service providers, international standard, 2015.

[IEC3-3] IEC 62443-3-3: Security for industrial automation and control systems – Part 3-3: System security requirements and security levels, international standard, 2015.

[ISO01] ISO/IEC 27001: Information Technology – Security Techniques – Information security management systems – Requirements, international standard, 2013.

[Ko16] Kobes, Pierre: Leitfaden Industrial Security, IEC 62443 einfach erklärt. VDE Verlag, 2016.

[NE-4d] NERC Standard CIP-006-4d – Cyber Security – Physical Security of Critical Cyber Assets, 2013.[VdS07] VdS 2007: Informationstechnologies (IT-Anlagen) – Gefahren und Schutzmaßnahmen, Publikation der deutschen Versicherer (GDV e.V.) zur Schadenverhütung, 2016.

[VdS33] VdS 2333: Sicherungsrichtlinien für Geschäft und Betriebe, VdS-Sicherungsrichtlinien, 2014.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 145

On the possible impact of security technology design on policy adherent user behavior: Results from a controlled empirical experiment

Sebastian Kurowski[1], Nicolas Fähnrich[2], Heiko Roßnagel[3]

Abstract: This contribution provides results from a controlled experiment on policy compliance in work environments with restrictive security technologies. The experimental setting involved subjects forming groups and required them to solve complex and creative tasks for virtual customers under increasing time pressure, while frustration and work impediment of the used security technology were measured. All subjects were briefed regarding existing security policies in the experiment setting, and the consequences of violating these policies, as well as the consequences for late delivery or failure to meet the quality criteria of the virtual customer. Policy breaches were observed late in the experiment, when time pressure was peaking. Subjects not only indicated maximum frustration, but also a strong and significant correlation (.765, p<.01) with work impediment caused by the security technology. This could indicate that user-centred design does not only contribute to the acceptance of a security technology, but may also be able to positively influence practical information security as a whole.

Keywords: Policy Compliance; Technology Acceptance; Task Technology Fit; Information Security; Due Care

1 Introduction

Understanding human adherence or deviance to information security policies is a key element for future security architectures. Human behaviour is an important antecedent for attacks on organizational and private information systems [Jo16], with 34.8% of german corporations reporting social engineering as a main cause of industry espionage [Co14], and human error being one of three root causes for data breaches [Po16]. Therefore users in information security are often treated as a potential vulnerability [AH09]. Existing literature indicates that users are either naïve poorly educated, or risk takers in their security behaviour [Ke17] and should be faced with due process (e.g. sanctions) [DHG08] in the case of non-compliance. Other research indicates that users are an important security asset and

[1] Fraunhofer-Institute for Industrial Engineering IAO, Competence Team Identity Management, Nobelstr. 12, 70569 Stuttgart, sebastian.kurowski@iao.fraunhofer.de
[2] University of Stuttgart, Institute for Labour Science and Technology Management IAT, Competence Team Identity Management, Nobelstr. 12, 70569 Stuttgart, nicolas.faehnrich@iat.uni-stuttgart.de
[3] Fraunhofer-Institute for Industrial Engineering IAO, Competence Team Identity Management, Nobelstr. 12, 70569 Stuttgart, heiko.rossnagel@iao.fraunhofer.de

doi:10.18420/sicherheit2018_11

should be considered more thoroughly [AS99, ZR12] in the design process of security architectures. This existing contradiction has led to an ongoing discussion in the topic of security technology design as to which extent a technology should meet user requirements [Fr07, ZR12, HRZ10], and to which extent a technology should enforce security aspects in order to provide a contribution to an effective security architecture. Finally, there are existing contributions that emphasize the impact of an individuals' environment, e.g. the actions of the individuals' peers [AH09] with regard to information security, or the impediment on the individuals work [KB13]. All these contribution have one thing in common: they mostly use static, momentary data capture methods such as self-reporting questionnaires, or focus on changes e.g. in password use, rather than observing human behaviour. This contribution aims at providing the question how human behaviour changes prior and after a policy violation. We aim to observe changes in social interactions, frustration, and use of security technologies in the context of a policy violation. In the following, we provide insights from a 2-day controlled experiment that was conducted in order to gain an understanding on the impact of the individuals' environment with regard to task load, social cues, and work impediment on the individuals' policy compliance. A security technology that provide strict policy enforcement capabilities was applied as a technical control in the setup. Since perfect policy enforcement is unrealistic in most real-world cases, the technical control was weakened with a backdoor. Use of this backdoor was clearly prohibited by an information security policy (administrative control) that was read to, handed out and signed by all participants. By observing the factors that occur during, prior and after a violation of the security policy by participants, we hope to get a more unified insight into what observable circumstances contribute to the security policy violation. The contribution is structured as follows: The following Section 2 provides an overview on currently used empirical research methodologies in security policy compliance research, and outlines why we have chosen a controlled experiment for our purposes. The experiment setup, sampling of participants, participant monitoring and measurement instruments are being introduced in Section 3. Observations regarding the policy compliance of the individuals, and a discussion of the observed factors that impact security policy compliance are laid out in Section 4, followed by conclusions in Section 5.

2 Methodology of Security Policy Compliance Research

Most of security policy compliance research uses self-reporting questionnaires, mostly implemented as web-surveys. For instance the contributions [ADO16, AMA15, AM14, BB13, BK07, BCB10, HB15, If16, KB13, Li14, PKS13, PH14, RFE16, Sa15, SKH15, WP13, WJS11, YBD16, YK13] use questionnaire item sets for measuring the intention to comply or the actual compliance. In this case individuals are asked, e.g. if they intend to comply with information security policies. Another possibility of measuring policy compliance or policy deviance is by using scenarios and asking the individual whether it would behave similarly as the individual described in the scenario. Such a factorial survey approach [RA82] is often used in research on policy deviant behaviour such as

[BS16, Ch13, DHS14, DHG08, Jo16] The advantages of both methods is obviously easier acquisition and maintenance of data, since the questionnaire must only be administered to the individuals, recollected and analysed. However, one must keep in mind that such surveys are often only able to provide a snapshot of policy compliance in time. Also in order for complex processes such as social interactions to be captured by these instruments, the researcher must anticipate these processes. In order to add context to the observation, and e.g. be able to find indications of not anticipated processes, one may refer to qualitative semi-structured interviews. Semi-structured interviews have e.g. been applied by [Ng09] in order to gain insights on the rather complex topic of security culture. Semi-structured interviews use pre-formulated questions, but do not require strict adherence to them [My09]. This adds the advantage that an interviewer is able to focus at the interview subjects' world, allowing improvisation and adaption of the interview process, while providing consistency between interviews. However, interviewers are not invisible to the interview subject, which may alter the situation and the outcome of the interview [FF05]. The contributions [SM16], [Je14], and [Va14] each use a controlled experiment. A controlled experiment allows for the observation of a group under treatment. The advantage of such a controlled experiment is that the condition of subjects can be observed in a controlled environment prior and after a treatment. This way changes in the conditions of the subjects cannot only be observed but also be accounted for. e.g. [Va14] used such a setup for observing the security behaviour of individuals. The research subject were observed in their behaviour on security warning disregard, while EEG data was being monitored. A security incident was applied as a treatment during the experiment and the experimenters could observe the changes in security perception. Obviously, there are methods available that involve easier maintenance and acquisition of data, such as self-reporting questionnaires, factorial survey methods, or qualitative semi-structured interviews. However, policy compliance may be subject to social cues, and environmental cues. Therefore, a controlled experiment, that allows the experimenters to measure and observe changes in social cues and subject behaviour, while controlling for changes in the subjects environment is applied for the research purposes of this contribution.

3 Methodology

3.1 Experiment Setup

The experiment setup aimed at providing a realistic engineering scenario, in which participants were equipped with an access control mechanism for use, while solving complex tasks for virtual customers. In their research on policy compliance in different professions, [Ra13] observe that innovative professions such as engineering or information systems showed the lowest indications for compliance with their organizations information security policy. Therefore, the experiment setup aimed at emulating engineering use cases. This was achieved by providing tasks that required the participants to innovate, be creative and improvise, all under increasing time pressure. Participants were required to use an

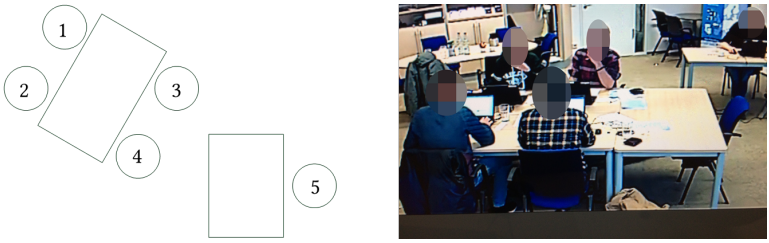


Fig. 1: Experiment setup. Participants were seated with respect to their role (left), whereas project leads (1-3) and the worker (4) were seated together, while the administrator (5) was seated apart. The right picture shows a screenshot from the observation stand of the setup.

access control mechanism. This mechanism was able to enforce access by encryption and signature of files. The mechanism was embedded into the computers operating system and could therefore be applied without any additional effort by the participants. The operating system used virtualization to provide Windows, Word and an E-Mail Client to the participants. Familiarity with Microsoft Windows and Word was a prerequisite during participant acquisition. Ten participants were randomly sampled into 2 groups from a sample of engineering, marketing and computer science students that had previously applied for participation in the experiment. The resulting groups however consisted of four students with a computer science background and one student with a marketing background in the first group, and 4 students with a computer science background and one student with an engineering background in the second group. Each group was participating for two days, whereas one day was set to 3 hours of experimentation, and the other day was set to 4 hours. Unfortunately, since the security mechanism was only available for a limited amount of time, scheduling of the experiment required one group to participate for 3 hours on the first and 4 hours on the second day, while the second group participated for 4 hours on the first, and 3 hours on the second day. This was solved by introducing a break each for the first and second group. This enabled both groups to stay aligned. The scenario, that participants were introduced into, looked as follows: The participants were told that they act as a virtual supplier in the automotive industry. They have three customers and the common duty to ensure due care, timely delivery and satisfaction of their customers quality expectations. This way, participants were able to receive a reward for their work between 80€ and 100€. However, if their group failed to deliver a task in due time, or to meet their customers quality expectations they would receive sanctions in terms of minus points. Each collected minus point collected by the group would then decrease the reward for each participant in the group. Additionally, participants were schooled that they were required to keep the data of their customers separate, as the customers themselves were competitors. This aligns well with the circumstances implied by engineering environments, such as in the automotive industry [WRZ12]. Participants were trained on how they could ensure this by using the access control mechanism provided to them. Participants were also told that failure to separate their customers data, may result in a security incident. Such an incident would also result in the group collecting a minus point. However, unlike in the

case of late delivery, or not satisfying their customers quality criteria, participants were also told that a security incident may only facilitate with a chance of approximately 2.8% (This is the probability of six eyes on two dices). Since individual misbehaviour does not with certainty but only by chance lead to a security incident, this further ensured realism of the scenario. The setup design used a backdoor: An e-mail client was provided to the participants for communication of the customers with the participants, and in order to inform the participants when measurements take place. Participants could also use this e-mail client for data exchange with each other. However, encryption of the exchanged data would not be provided via e-mail. Participants were thus told, that any exchange of customer data between each other would be an unsecure data exchange and could thus lead to a security incident with a chance of approximately 2.8%, and thus to a minus point for the whole group. Instead, participants should exchange data via secured USB sticks which, while requiring more effort, enabled them to apply the access control mechanism on the data stored on the USB stick. Participants were not only schooled regarding the adequate security behaviour, but were also provided with this information in a written policy. While all participants shared the common duty for due care, timely delivery and satisfaction of the customers quality criteria, some participants had additional roles. Three participants were assigned as project leads, one project lead for each customer. The special responsibility of the project leads was to communicate with the customer, which included receiving the work specification from and providing the work result to the customer. They also had the responsibility to ensure secure, timely and adequate project execution. The other special role that was assigned was the administrator. The access control mechanism that was applied by all participants separated data between the projects for the different virtual customers. This was done by providing a group-like concept for the virtual customers. The administrators role was to ensure assignment and revocation of participants to the different groups. Its special duty was to ensure that no access rights creep was taking place so that the need-to-know was ensured at each participant. The administrator could do this by communicating with a virtual IT, demanding the assignment or revocation of participants to groups. Apart from these special roles, all participants also had the duty to work on the projects (worker role). Figure 1 provides an overview on the setup. Participants were told about the different special roles, and all participants were told to assign themselves to a special role if desired. Participants were seated, with regard to their role. This ensured that the participant observation could be more easily attributed to the participant role. All participants were briefed prior to the experiment regarding the increase in stress over time, but not regarding the hypotheses and objectives of the experiment. Particpants were also trained and reminded that they are able to cancel the experiment at any time. However, participants were told, why the stress was increased by the different suppliers after the experiment. This debriefing also included a detailed explanation of the experiment setup, the hypotheses and the overall objective of the experiment. All participants were observed during the experiment, in order to be able to cancel the experiment if any harm or danger is imminent for the individuals.

3.2 Treatments

Over the time of the experiment, the work load of the participants was gradually increased. The goal was to see, what amount of stress level will be measurable when the participants break the policy. Increase of stress was achieved by the projects that the participants were required to solve. These treatments were gradually, increased by decreasing (a) available time resources by decreasing the project deadlines, and (b) available work force by parallel customer requests. An example for such a treatment is provided in the Annex. The treatments

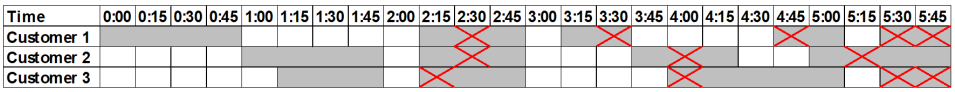


Fig. 2: Gantt chart of the treatments relative to the experiment time. Project-based treatments are shown in gray, whereas the right end of each gray area indicates the project deadline. Red crosses indicate treatments through customer interaction..

over time are shown in Figure 2. The experiment started with only customer 1 asking for delivery, allowing participants to work collaboratively. However, until hour 3:00 into the experiment, the workforce is first decreased from 1:15 to 2:00, and then the workforce is decreased along with the available time resources by requiring the participants to provide results to all three customers in less time than in the previous projects. The treatment then drastically decreases the available time resources by requiring delivery for a project between 3:15 and 3:45. And then goes on into further decreasing the workload. The overall goal of this schedule is to continuously increase the amount of workload that the participants receive, by decreasing available time and workforce. Additionally, treatments were provided through customer interactions. If participants have already become familiar with the work, these interactions were designed to increase uncertainty and thus the participants stress. Such customer interactions could be complaints about the quality and threats of minus points by the customers, demanding additional work for projects shortly before the deadline, requiring an earlier delivery of the project results, demanding a draft of the work results or even a final draft with 90% completion status, or asking the project lead who currently works on the project. For instance, one customer might ask after 50% of the projects' time has passed, who is working on the project at the supplier. After the suppliers response, the customer would then ask for an earlier delivery. This combination of treatments by time reduction and work force reduction through the project situation, and increasing uncertainty at the participants by customer interaction, both aimed at gradually increasing the workload of the participants.

3.3 Data Acquisition

The experiment subjects were treated in an isolated room, in which they could act and organize autonomously. The room was monitored via three cameras that recorded and allowed the experiment team to observe the participants behaviour. The experiment team was

situated in a separate room in order to minimize interference of the team with the subjects. In fact, after the participants were introduced, their consensus gathered, and led into the experiment room, the door was closed, and no interference from outside the experiment room was permitted. The experiment team consisted of two researchers, who created a protocol of the participants actions and group behaviour. By monitoring the mail server that was setup for the experiment, the experiment team also observed if any policy violations took place. During the experiment participants would fill out seven surveys at predefined times into the experiment. These surveys consisted of an item set for measuring work impediment by the security technology, taken from [KB13], along with a paper version the NASA Task Load Index (TLX) measurement instrument [Ha06]. The latter provides participants with the possibility to indicate their perceived mental, physical, and temporal demand by the task, their performance, effort and frustration. Since NASA TLX is a subjective measure of workload, results may vary drastically between participants. Therefore weighting of the participants indications is required. In order to provide this weighting, participants were required to answer a survey, which weighted the seven different categories of workload to each other. This resulted in an individual weighting of the different categories for each participant. The combination of experimenter protocols, the Work Impediment Item set, and NASA TLX allows both for capturing the participants work load and stress, the perceived work impediment by the security technology and the participants' and participant group behaviour. We assume that measurements of the workload that is created by working on an objective is virtually indistinguishable from the workload that is created by working on an objective with a certain security mechanism. Therefore, work impediment introduces a scale that focuses only on the impediment created by a security mechanism when working on an objective.

4 Analysis and Discussion

4.1 Observation of the groups

In the beginning (t+0:00) of the first day, both groups started off with the lowest amount of pressure, at least according to the treatments (see Section 3.2). It was observed, that the subjects of the first group had trouble understanding the provided access control system. Especially, the concept of groups seemed confusing in the beginning. This caused the subjects to work around the technology and to work without computers by dictating to the project lead, what he or she should write into the result report for the customer. After the break of the first group, the observed frustration seemed to increase, while three parallel projects were requested by the customers. Now the subjects seemed to arrange into smaller groups, and information exchanges via USB stick and in line with the communicated policy took place. During the experiment it became apparent, that the first group, even though they showed technical expertise, had a hard time to understand the security mechanism and how to adjust the group memberships. Still, even when, later in the experiment day, the USB sticks malfunctioned, the subjects remained to work in line with the policy. The second

group showed more technical interest and competence than the first group. Already in the first project, this group build a hierarchy, whereas the administrator seemed to assign work and check-off decisions made by the group. This resulted in work being shared already in the first project. During the second project however, the group already started discussing whether they should violate the policy, since it would be quicker and deemed to be unlikely that something happens. However, the administrator as the informal leader of the group vetoed this option, which led to the other participants immediately disregarding the policy violation. Data exchange happened via USB stick, in compliance with the security policy. After the end of the first day both groups said they felt that the security mechanism restricted their work. The mechanism automatically encrypted every file. This effect was not visible for the participants, when working in one group. However, when transferring data from one group to another, e.g. via clipboard, participants only received encrypted data. While this strict enforcement of data separation between the groups was meant to be, it became evident that when participants were accidentally researching, e.g. with a web browser, for one customer but in the group of another customer, this feature would start to become annoying and obstructive for them. On the second day, the second group showed that they took the separation of customer data very seriously. Even though the time pressure was increased and the available work force was decreased due to parallel running projects, policy violations at first did not take place. However, 1:39 hours into the second day (at t+3:59), the USB sticks started to malfunction for the second group, and they started to violate the security policy after it became apparent to the group that they could not bypass this issue by e.g. dictating contents to the project lead who compiled the project report. Frustration and resignation of the participants become visible after this point, and more and more policy violations took place with the justification that nothing had yet happened (2:20 hours into the day at t+4:35 and 2:39 hours into the day at t+4:54). The first group made more use of the possibility to assign and revoke users to customer groups on the second day. This resulted in more cooperation between the participants. Information was being exchanged with USB sticks. However, it seemed to confuse participants sometimes that content that was not assigned to the group of their customer was encrypted. Finally, with increasing stress and being shortly before the deadline of a project, one of the project leads started to show a maximum amount of stress, yelling to another cooperating participant to “send the [. . .] thing per e-mail!”. This policy violation took place 1:47 hours into the second experiment day at t+5:02. This means, that while the policy violation in the second group took place at a later point in the experiment itself, it took place at a similar point in time of the experiment day for both groups (1:39 and 1:47 hours after start of the project), and at the moment of maximum visible frustration and highest time pressure. The experiment ended after t+06:00 hours, which means that the policy violation for the second group took place nearly at the end of the whole experiment. Both groups seemed to be very exhausted at the end of the experiment.

4.2 Task Load Index and Work Impediment

Overall, seven measurements of task load and work impediment were obtained per participant. The NASA TLX values were then weighted for each participant, by using the weights obtained from the survey at the end of the experiment (see Section 3.3). Since the experiment was spread over two days, participants had time to recreate between measurement three and four for group 1, and measurement two and three for group 2. Although, breaks were included in the experiment, to provide each group also with a moment to recreate, TLX values may be biased by this circumstance. Therefore the TLX values were further normalized with the minimum and maximum value for each participant on the respective experiment day. The resulting TLX values are provided in the annex.

The TLX values for both groups show, that the largest changes seem to be indicated with Frustration, Performance and Effort. Temporal Demands, while increasing, seem to only play a minor role. Cognitive demands remain relatively untouched by the treatments. Finally, physical demands as expected were all zero, since the factor was weighted to zero by all participants. Therefore physical demands are excluded from our analysis. In order to analyse the TLX values, we chose a correlation analysis. The results are indicated in Table 1[4] . It is clearly visible that the treatments resulted in an increase of the perceived performance (A self assessment of how successful the participant has been), , the participants effort (how hard was the work for the participant?), and the participants frustration. The indicated correlations are all highly significant ($p \leq .01$) and indicate high values. Temporal demands only weakly correlate with the participants frustration ($p \leq .05$). Also interesting is the relationship between work impediment, frustration, performance, and effort. The larger the demand that was introduced by the treatements showed to be, the larger the values for frustration, performance, effort and work impediment become. Illustrative for both experiment groups, showing this correlation are provided in the annex. The correlation between frustration and work impediment becomes evident in both, and is also indicated by the correlation analysis ($.765, p \leq .01$). Also the perceived work impediment by the security technology rises with the perceived performance ($.775, p \leq .01$), and the perceived effort ($.705, p \leq .01$).

4.3 Discussion of the findings

The participants, though affine to technology, were all neither security experts, nor familiar with security technologies. Yet, on the first day of the experiment one group decided to rather work with obstacles (e.g. by dictating contents to the project lead), than breaking the

[4] Due to length restrictions the annex to this contribution, that contains the data visualization and data tables along with the questionnaire, was not included in the printed version. It is however available online at: https://www.researchgate.net/profile/Sebastian_Kurowski/publication/323177670_Annex_-_On_the_possible_impact_of_security_technology_design_on_policy_adherent_user_behavior_Results_from_a_controlled_experiment/data/5a8479d4a6fdcc201b9ef0eb/sicherheit2018-annex.pdf

policy. The other group, although considering policy violations, kept to the security policy. Policy violations were, in both groups, only observed late into the experiment. The second group violated the policy twice between the fifth and sixth measurement, and again after the sixth measurement. The first group on the other hand took until shortly after the sixth measurement to violate the policy. This was surprising, since obviously all participants knew that there is only a minor chance that they got a disadvantage out of the action and it would have improved their work efficiency and reduced their level of task load. Participants argued after the end of the first day of the experiment that they wanted to keep the customer data secure, although they had no security expertise whatsoever. Another interesting observation is the correlation of work impediment. We would have expected, that work impediment remains steady, or at most weakly correlates with the amount of effort, performance and frustration. However, work impediment indicates one of the strongest correlations with these task load factors. Also in both groups, the policy violation could be observed at 1:39 / 1:47 and at a point when the frustration and work impediment each climaxed (between the fifth and the sixth measurement). This of course raises the suspicion, that the surge in frustration, work load and work impediment may be a cause for the policy violation. Participants all had one thing in common: they complained about the usability of the security technology. This is also shown by the work impediment. Furthermore, the bad user experience seemed to contribute to the frustration of the participants. The Theory of Planned Behaviour (TPB) [AH09], that plays a role in some contributions for information security policy compliance [SKH15, Hu12, If12, BCB10] may provide a possible explanation for this observation. TPB postulates that human behaviour is constituted out of the attitude of the individual towards the behaviour, the subjective norm of the behaviour, and the perceived behavioural control of the individual over the behaviour. The subjective norm hereby indicates the amount of pressure implied by the individuals peers. Since a violation of the security policy would have led to the chance of sanctions for the whole group, each participant may have perceived the subjective norm of not violating the security policy. The attitude of the individual may also have been positive towards keeping the customer data safe. This also aligns well with the observations from Section 4.1. However with increasing reduction of available time, work-force and the increase of uncertainty came an increase not only in performance and required effort, but also in the work impediment. This means, that the lack of usability was perceived worse, when under more stress by the individuals. This may have led to a change in attitude, from keeping the customer data secure towards not failing to meet the deadline and risking certain minus points. This is obviously a blatant contradiction to the naïve, unknowing or risk-affine user. None of our participants exhibited naivity or risk-affinity. All were unknowing. But neither violated the policy in the beginning. However, our results indicate that the work conditions, including the design of the security technology, may actually contribute to a security policy violation, not the disposition of the user.

5 Conclusion

To our knowledge this is the first time such a controlled experiment has been conducted in the field of security policy compliance research. The advantage of our approach over classic survey-based approaches that are widely used in policy compliance research lie in the opportunity to observe changes to behaviour prior and after the policy violation and to observe the actions that led to the policy violation. Our measurements indicated, that the work impediment of the security technology is perceived differently under different workloads, and that a surge in work impediment, frustration, performance and effort of the tasks contribute to a policy violations. Our observations showed that the participants, albeit being unknowing of security, showed a large willingness to protect the customer data and even switched to more work-intensive workarounds even though they could have gained an obvious advantage from breaking the policy. An explanation for this observation however could be provided by a change of attitude in the light of TPB. If these observations are true, this means that the users role in policy compliance must be reconsidered. Policy violations may be subject to bad working conditions, and the security technology design. All these correlations show a high likeliness of not being random ($p \leq .01$) even though rather small sample sizes were used in the experiment. We therefore argue that the impact of the security technology, along with basic assumptions about the user in information security must be reconsidered in order to provide an effective and secure working environment.

References

[ADO16] Abed, J.a; Dhillon, G.a; Ozkan, S.b: Investigating continuous security compliance behavior: Insights from information systems continuance model. In: AMCIS 2016: Surfing the IT Innovation Wave - 22nd Americas Conference on Information Systems. 2016.

[AH09] Albrechtsen, E.; Hovden, J.: The information security digital divide between information security managers and users. Computers & Security, 28(6):476–490, September 2009.

[AM14] Aurigemma, S.a; Mattson, T.b: Do it OR ELSE! exploring the effectiveness of deterrence on employee compliance with information security policies. In: 20th Americas Conference on Information Systems, AMCIS 2014. 2014.

[AMA15] Al-Mukahal, H.M.a; Alshare, K.b: An examination of factors that influence the number of information security policy violations in Qatari organizations. Information and Computer Security, 23(1):102–118, 2015.

[AS99] Adams, Anne; Sasse, Martina Angela: Users are not the enemy. Commun. ACM, 42(12):40–46, December 1999.

[BB13] Borena, B.a; Bélanger, F.b: Religiosity and information security policy compliance. In: 19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime. volume 4, pp. 2848–2855, 2013.

[BCB10] Bulgurcu, B.; Cavusoglu, H.; Benbasat, I.: Information security policy compliance: An empirical study of rationality-based beliefs and information security awareness. MIS Quarterly: Management Information Systems, 34(SPEC. ISSUE 3):523–548, 2010.

[BK07] Boss, S.R.a; Kirsch, L.J.b: The last line of defense: Motivating employees to follow corporate security guidelines. In: ICIS 2007 Proceedings - Twenty Eighth International Conference on Information Systems. 2007.

[BS16] Bansal, G.; Shin, S.I.: Interaction effect of gender and neutralization techniques on information security policy compliance: An ethical perspective. In: AMCIS 2016: Surfing the IT Innovation Wave - 22nd Americas Conference on Information Systems. 2016.

[Ch13] Cheng, L.a; Li, Y.a b; Li, W.a; Holm, E.c; Zhai, Q.c: Understanding the violation of IS security policy in organizations: An integrated model based on social control and deterrence theory. Computers and Security, 39(PART B):447–459, 2013.

[Co14] Corporate Trust: Studie: Industriespionage 2014 - Cybergeddon der deutschen Wirtschaft durch NSA & Co.? Studie, Coprorate Trust Business Risk & Crisis Management GmbH, München, 2014.

[DHG08] D'Arcy, J.; Hovav, A.; Galleta, D.: User Awareness of Security Countermeasures and Its Impact on Information Systems Misuse: A Deterrence Approach. Information Systems Research, pp. 1–20, 2008.

[DHS14] D'Arcy, J.a; Herath, T.b; Shoss, M.K.c: Understanding Employee Responses to Stressful Information Security Requirements: A Coping Perspective. Journal of Management Information Systems, 31(2):285–318, 2014.

[FF05] Frey, James H; Fontana, A: The interview: From neutral stance to political involvement. The Sage handbook of qualitative research, pp. 695–726, 2005.

[Fr07] Fritsch, Lothar: Privacy-Respecting Location-Based Service Infrastructures: A Socio-Technical Approach to Requirements Engineering. Journal of Theoretical and Applied Electronic Commerce Research, 2(3), 2007.

[Ha06] Hart, Sandra G: NASA-task load index (NASA-TLX); 20 years later. In: Proceedings of the human factors and ergonomics society annual meeting. volume 50. Sage Publications Sage CA: Los Angeles, CA, pp. 904–908, 2006.

[HB15] Humaidi, N.a; Balakrishnan, V.b: The Moderating effect of working experience on health information system security policies compliance behaviour. Malaysian Journal of Computer Science, 28(2):70–92, 2015.

[HRZ10] Hühnlein, D.; Roßnagel, H.; Zibuschka, J.: Diffusion of Federated Identity Management. In (Freiling, F.C., ed.): Sicherheit 2010, pp. 25–36. Köllen Druck + Verlag GmbH, Bonn, 2010.

[Hu12] Hu, Q.a; Dinev, T.b; Hart, P.b; Cooke, D.c: Managing Employee Compliance with Information Security Policies: The Critical Role of Top Management and Organizational Culture. Decision Sciences, 43(4):615–660, 2012.

[If12] Ifinedo, P.: Understanding information systems security policy compliance: An integration of the theory of planned behavior and the protection motivation theory. Computers and Security, 31(1):83–95, 2012.

[If16] Ifinedo, P.: Critical Times for Organizations: What Should Be Done to Curb Workers' Noncompliance With IS Security Policy Guidelines? Information Systems Management, 33(1):30–41, 2016.

[Je14] Jenkins, J.L.a; Grimes, M.b; Proudfoot, J.G.b; Lowry, P.B.c: Improving Password Cybersecurity Through Inexpensive and Minimally Invasive Means: Detecting and Deterring Password Reuse Through Keystroke-Dynamics Monitoring and Just-in-Time Fear Appeals. Information Technology for Development, 20(2):196–213, 2014.

[Jo16] Johnston, A.C.a; Warkentin, M.b; McBride, M.c; Carter, L.d: Dispositional and situational factors: Influences on information security policy violations. European Journal of Information Systems, 25(3):231–251, 2016.

[KB13] Kajtazi, M.a; Bulgurcu, B.b: Information security policy compliance: An empirical study on escalation of commitment. In: 19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime. volume 3, pp. 2011–2020, 2013.

[Ke17] Kelm, D.: Security-Fatigue I - Wenn Anwender es müde sind, sich um Sicherheit zu bemühen - und was man dagegen tun kann. <kes> Die Zeitschrift für Informations-Sicherheit, 33(4):54–57, August 2017.

[Li14] Li, H.a; Sarathy, R.b; Zhang, J.c; Luo, X.d: Exploring the effects of organizational justice, personal ethics and sanction on internet use policy compliance. Information Systems Journal, 24(6):479–502, 2014.

[My09] Myers, MD: Qualitative research in business and management. Sage Publications Ltd, London, 1 edition, 2009.

[Ng09] Ngo, Leanne; Zhou, Wanlei; Chonka, Ashley; Singh, Jaipal: Assessing the level of I.T. security culture improvement: Results from three Australian SMEs. IEEE, pp. 3189–3195, November 2009.

[PH14] Putri, F.; Hovav, A.: Employees' compliance with BYOD security policy: Insights from reactance, organizational justice, and protection motivation theory. In: ECIS 2014 Proceedings - 22nd European Conference on Information Systems. 2014.

[PKS13] Pahnila, S.a; Karjalainen, M.a; Siponen, M.b: Information security behavior: Towards multistage models. In: Proceedings - Pacific Asia Conference on Information Systems, PACIS 2013. 2013.

[Po16] Ponemon Institute: 2016 Cost of Data Breach Study: Global Analysis. Benchmark research sponsored by IBM, Ponemon Institute, IBM, Traverse City, Michigan, USA, June 2016.

[RA82] Rossi, Peter H; Anderson, Andy B: The factorial survey approach: An introduction. Measuring social judgments: The factorial survey approach, pp. 15–67, 1982.

[Ra13] Ramachandran, Sriraman; Rao, V Srinivasan Chino; Goles, Timothy; Dhillon, Gurpreet: Variations in information security cultures across professions: a qualitative study. Communications of the Association for Information Systems, 33(11):163–204, 2013.

[RFE16] Rocha Flores, W.; Ekstedt, M.: Shaping intention to resist social engineering through transformational leadership, information security culture and awareness. Computers and Security, 59:26–44, 2016.

[Sa15] Safa, N.S.a; Sookhak, M.a; Von Solms, R.b; Furnell, S.c; Ghani, N.A.a; Herawan, T.a: Information security conscious care behaviour formation in organizations. Computers and Security, 53:65–78, 2015.

[SKH15] Sommestad, T.; Karlzén, H.; Hallberg, J.: The sufficiency of the theory of planned behavior for explaining information security policy compliance. Information and Computer Security, 23(2):200–217, 2015.

[SM16] Shepherd, M.M.a; Mejias, R.J.b: Nontechnical Deterrence Effects of Mild and Severe Internet Use Policy Reminders in Reducing Employee Internet Abuse. International Journal of Human-Computer Interaction, 32(7):557–567, 2016.

[Va14] Vance, A.a; Eargle, D.b; Anderson, B.B.a; Brock Kirwan, C.a: Using measures of risk perception to predict information security behavior: Insights from electroencephalography (EEG). Journal of the Association of Information Systems, 15:679–722, 2014.

[WJS11] Warkentin, M.a; Johnston, A.C.b; Shropshire, J.c: The influence of the informal social learning environment on information privacy policy compliance efficacy and intention. European Journal of Information Systems, 20(3):267–284, 2011.

[WP13] Wall, J.D.; Palvia, P.: Control-related motivations and information security policy compliance: The effect of reflective and reactive autonomy. In: 19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime. volume 2, pp. 894–902, 2013.

[WRZ12] Wehrenberg, Immo; Roßnagel, Heiko; Zibuschka, Jan: Secure Identities for Engineering Collaboration in the Automotive Industry. Bamberg, pp. 202–213, 2012.

[YBD16] Yaokumah, W.a; Brown, S.b; Dawson, A.A.c: Towards modelling the impact of security policy on compliance. Journal of Information Technology Research, 9(2):1–16, 2016.

[YK13] Yoon, C.; Kim, H.: Understanding computer security behavioral intention in the workplace: An empirical study of Korean firms. Information Technology and People, 26(4):401–419, 2013.

[ZR12] Zibuschka, Jan; Roßnagel, Heiko: On Some Conjectures in IT Security: The Case for Viable Security Solutions. Presented at the SICHERHEIT 2012. 2012.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 159

Towards Forensic Exploitation of 3-D Lighting Environments in Practice

Julian Seuffert[1], Marc Stamminger[2], Christian Riess[3]

Abstract: The goal of image forensics is to determine authenticity and origin of a digital image or video without an embedded security scheme. Among the existing methods, the probably most well-known physics-based approach is to validate the distribution of incident light on objects of interest. Inconsistent lighting environments are considered as an indication of image splicing. However, one drawback of this approach is that it is quite challenging to use it in practice.

In this work, we propose several practical improvements to this approach. First, we propose a new way of comparing lighting environments. Second, we present a factorization of the overall error into its individual contributions, which shows that the biggest error source are incorrect geometric fits. Third, we propose a confidence score that is trained from the results of an actual implementation. The confidence score allows to define an implementation- and problem-specific threshold for the consistency of two lighting environments.

Keywords: multimedia security; image forensics; physics-based forensics; 3-D lighting estimation

1 Introduction

Visual media plays an important role in our everyday communication. This is partly due to the widespread availability of consumer cameras, and partly due to the ease of distributing visual media, for example over social media. These new possibilities allow to document events in an unprecedented density. However, when a picture or video is taken as evidence that an event actually happened, it is also important to be able to verify its authenticity. Blind image forensics aims to provide technical tools to authenticate visual media without the help of an embedded security scheme. In the recent years, several books and overview papers have been published on image forensics, for example [RTD11, SM13, Fa16].

This work is about so-called physics-based methods in image forensics. The guiding idea of these methods is to validate the physics in the depicted scene for its consistency,

[1] Friedrich-Alexander University Erlangen-Nuernberg, IT-Security Infrastructures Lab, Martensstr. 3, 91058 Erlangen, Germany julian.seuffert@fau.de
[2] Friedrich-Alexander University Erlangen-Nuernberg, Computer Graphics Lab, Cauerstr. 11, 91058 Erlangen, Germany marc.stamminger@fau.de
[3] Friedrich-Alexander University Erlangen-Nuernberg, IT-Security Infrastructures Lab, Martensstr. 3, 91058 Erlangen, Germany christian.riess@fau.de

doi:10.18420/sicherheit2018_12

like the direction and color of the incident light [JF07a, RA10], consistency of specular highlights [JF07b], reflections [OF12], or shadows [Zh09, KOF14]. However, current physics-based methods have the disadvantages that their applicability depends on the visual content in the scene, and that they typically require user interaction.

Several works perform the analysis to human faces only [KF10, Pe16, Pe17]. Faces commonly occur in pictures and videos, and it may be a forensic goal to validate the composition of people in a scene. Operating on faces has the advantage that there exists software to estimate a 3-D model of the face, which then allows to compute the 3-D distribution of incident light.

Nevertheless, using 3-D lighting forensics in practice is still challenging, and oftentimes requires expert knowledge. The goal of this paper is to narrow the gap from the base algorithm in literature towards its use in practice. Specifically, we present an systematic analysis of the algorithmic steps, and we propose a practical, trainable confidence score that adapts to the specific implementation of the algorithm at hand. Using the confidence score, a practitioner not only obtains an assessment whether two lighting environments are identical, but also a probability with which this assessment is true. In detail, the contributions of the paper consist of

- A study of the spherical harmonics model for lighting representation and comparison with an approach that avoids extrapolation over surface normals without observations.
- A factorization of the fitting error into its individual physical contributions, to better understand its impact on the estimation.
- A proposal for a confidence score that describes the reliability of the individual pipeline implementation.

The paper is organized as follows. We present the underlying model and the computation of lighting environments in Sec. 2. Approaches for comparing lighting environment are presented in Sec. 3, the error factorization in Sec. 4, and the proposed confidence score in Sec. 5. Finally, our experiments are presented in Sec. 6.

2 3-D Lighting Estimation

Johnson, Kee, and Farid [JF07a, KF10] proposed to estimate the distribution of incident light on objects of interest in a scene. Such a distribution is called a lighting environment. Assuming a single, infinitely distant light source (such as the sun in outdoor scenes) and no inter-reflections, then all scene objects must exhibit identical lighting environments. With minor modifications, these assumptions can be used to obtain a forensic test on a given input image, by approximating the rays of the sun or another distant light source as parallel.

Johnson and Farid proposed to compute 2-D lighting environments from monochromatic object contours. Kee and Farid proposed later to estimate more reliable 3-D lighting

environments from the shape of human faces [KF10], using existing software for fitting a 3-D face shape to a 2-D image.

Kee and Farid model a lighting environment with the help of surface normals and their associated pixel intensities [KF10]:

$$e(\vec{n}) = \int_{\Omega} l(\vec{v}) r(\vec{v}, \vec{n}) d\Omega \ . \tag{1}$$

Here, $e(\vec{n})$ is the observed intensity of a pixel with surface normal $\vec{n}$. Ω is the hemisphere of all light directions that fall on a patch with surface normal $\vec{n}$, $l(\vec{v})$ denotes light that falls onto that pixel coming from direction $\vec{v}$, and $r(\vec{v}, \vec{n})$ denotes the reflectance function for that patch. The model becomes particularly convenient when $r(\vec{v}, \vec{n})$ is assumed to be Lambertian (purely diffuse), such that the reflected intensity is the cosine between $\vec{v}$ and $\vec{n}$.

Under the assumption of Lambertian reflectance, the observed intensity $e(\vec{n})$ can be represented by second order spherical harmonics, i.e.,

$$e(\vec{n}) = \sum_{n=0}^{2} \sum_{m=-n}^{n} l_{n,m} Y_{n,m}(\vec{n}) \ , \tag{2}$$

where $Y_{n,m}$ denotes the m-th spherical harmonics basis function of order n, and $l_{n,m}$ is a weighting coefficient. These coefficients can be directly estimated from the observed intensities. Let $i(\vec{x}_i)$ denote the intensity of the i-th pixel $\vec{x}_i$ from a face, and $\vec{n}(\vec{x}_i)$ the surface normal of the face at position $\vec{x}_i$. Then, Kee and Farid propose to estimate the lighting coefficients $\vec{l}$ from the linear equation

$$\mathbf{M} \cdot \vec{l} = \vec{b} \ , \tag{3}$$

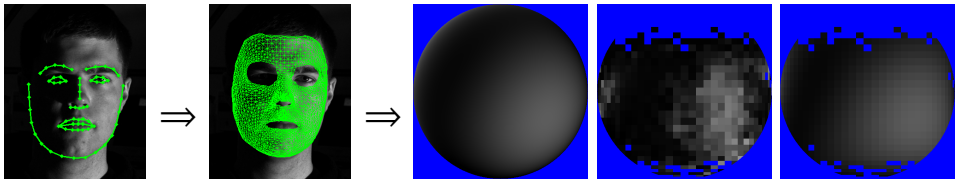
where $\vec{b} = (i(\vec{x}_1), \ldots, i(\vec{x}_N))^{\mathrm{T}}$ are the observed intensities and

$$\mathbf{M} = \begin{pmatrix} \pi Y_{0,0}(\vec{n}(\vec{x}_1)) & \ldots & \frac{\pi}{4} Y_{2,2}(\vec{n}(\vec{x}_1)) \\ \vdots & \ddots & \vdots \\ \pi Y_{0,0}(\vec{n}(\vec{x}_N)) & \ldots & \frac{\pi}{4} Y_{2,2}(\vec{n}(\vec{x}_N)) \end{pmatrix} \ . \tag{4}$$

Equation 3 is solved for $\vec{l}$ via least squares, i.e.,

$$\vec{l} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\vec{b} \ . \tag{5}$$

The mathematical framework by Kee and Farid is elegant, but several special cases are not explicitly discussed. For example, Peng *et al.* later proposed to automate this pipeline [Pe16], and to add a more complex model for surface reflectance [Pe17].



(a) extract landmarks (b) custom. face model (c) spherical harmonics (d) IBS (e) HIBS

Fig. 1: Computed landmarks (a) and fitted 3-D model (b) for obtaining per pixel a 3-D surface normal. Lighting environments can be computed as spherical harmonics [JF07a, KF10] (c), intensity binned spheres (d), or hybrid intensity binned spheres (e).

3 Comparison of Lighting Environments

Spherical harmonics are the standard representation for lighting environments [JF07a, KF10]. Two lighting environments are compared by expanding the spherical harmonics coefficients $\vec{l}$ of each object of interest to an intensity profile. The correlation between two such intensity profiles is then used to quantify their similarity.

The suitability of spherical harmonics for forensic lighting estimation has to our knowledge not been investigated. We find this surprising, as spherical harmonics consist of low frequencies averaged over potentially unevenly distributed or missing observations, which might lead to wrong results: for example, persons with dense head hair will contribute almost no surface normals pointing upwards. Nevertheless, the spherical harmonics model represents and weighs all angular directions of the hemisphere pointing towards the camera equally. Thus, it is not possible to distinguish the impact of the actual observations on similarity ρ from artifacts from extrapolation of lighting environments. An example spherical harmonics model is shown in Fig. 1c. To compare two spherical harmonics representations, we compute their correlation directly on their SH coefficients as proposed earlier [JF07a].

As an alternative, we propose to consider what we call a "intensity binning sphere" (IBS), where intensities are binned by their surface normals, in steps of 5°. A concrete example IBS is shown in Fig. 1d. In contrast to Fig. 1c, the model does for example not cover surface normals that point upwards, as we do not observe any skin area with that orientation. This idea can be further improved by combining both approaches into a representation that we call "hybrid IBS" (HIBS), shown in Fig. 1e. Here, the spherical harmonics model is cropped to angular bins with a width of 5° that are actually filled with observations. Two IBS or HIBS representations are compared by computing the correlation over the intersection of non-empty bins of both representations.

(a) diff. & spec. (b) purely diffuse (c) global lighting (d) local lighting

Fig. 2: Example renderings from the synthetic dataset. From left to right: diffuse and specular, purely diffuse, purely diffuse with global lighting, purely diffuse with local lighting.

4 Error Factorization

Errors of the individual methods accumulate in the processing chain. To optimize the algorithm, it is important to understand the relative contribution of each factor to the overall error. Based on the physical model in Eqn. 1, we investigate three potential sources of error. First, the face fit might yield slightly incorrect surface normals. Second, the required reflectance model might be more complex than the assumed pure Lambertian model. Third, self-shadows due to occlusions could have an impact on real images.

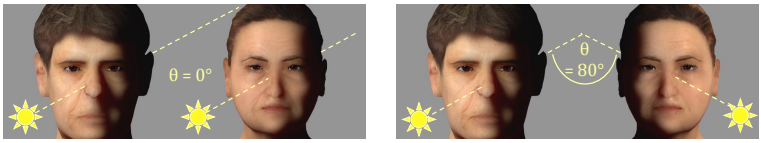
To understand the impact of these three factors, we created a synthetic face dataset consisting of 12 subjects using *MakeHuman* and *Blender*. In *MakeHuman*, subjects are created from a parameters like age or gender. The surface normals from the synthetic face model can be directly taken as ground truth to evaluate the estimation errors in the geometry.

To obtain data for the other two hypotheses, we re-rendered the data with the default amount of specular reflections and the computation of self-shadows. Example faces are shown in Fig. 2. Local lighting denotes illumination of surface patches without self-shadowing. Global lighting includes self-shadowing. The limited photo realism in this rendered data is a minor concern, as only relative performance differences between the variants are considered. The dataset and the code for generating the data is publicly available on our website[4].

5 Confidence Score

We seek to transform the correlations from Sec. 3 into a confidence score that indicates whether the underlying lighting environments might be identical. Computation of the confidence score depends on the concrete implementation of the face fitting and lighting environment computation. Thus, we propose to learn it from training data, i.e., from face images that are acquired under known lighting. In our experiments, we further assume a single dominant light source, such that we can determine the angular resolution of the lighting environment estimation. Let us denote identical lighting environments as “consistent”, and different lighting environments as “inconsistent”. By extension, we consider faces under

[4] https://faui1-files.cs.fau.de/public/mmsec/datasets/sfd



(a) consistent lighting (b) inconsistent lighting

Fig. 3: Angle θ between the dominant light sources on two faces.

identical lighting as consistent, and faces under different lighting as inconsistent. If only a single light source illuminates a face, let θ be the angular difference between the dominant light sources of both faces. Then, as shown in Fig. 3, $\theta \neq 0$ indicates inconsistent lighting environments (positive class), and $\theta = 0$ consistent lighting environments (negative class).

We consider the confidence score as a function of the type I and type II error on the respective dataset. The type I error α denotes the relative number of samples being incorrectly classified as inconsistent. The type II error β denotes the relative number of samples being incorrectly classified as consistent. $\kappa_p = 1 - \alpha$ and $\kappa_n = 1 - \beta$ denote the confidence score of labelling the lighting as "inconsistent" and "consistent", respectively.

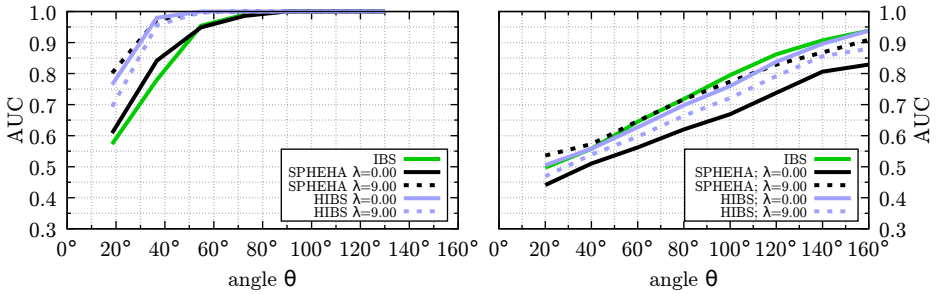
One possible decision template is to choose the class with the highest confidence score κ_n or κ_p. Hence, such a decision exhibits a confidence score function $\kappa_{np}(\rho) = \max(\kappa_n(\rho), \kappa_p(\rho))$ whereas $\kappa_n = 1 - \kappa_p$.

6 Experiments

We first describe the used datasets and experimental protocol in Sec. 6.1 and Sec. 6.2, and then present the experimental results in Sec. 6.3.

6.1 Datasets

We model 12 synthetic subjects, consisting of two males and two females of African, Asian and Caucasian descent and varying age using *MakeHuman v1.1.1*. Rendering is performed with *Blender v2.79*. We refer to this dataset as the Synthetic Face Dataset (SFD). Each subject is illuminated by nine distant point lights with a pitch angle of $\phi = 23°$ and with a yaw angle of $\psi \in \{-80°, -60°, \ldots, +80°\}$. Two light sources exhibit an angular distance of $\theta \in \Theta_{\text{SFD}} = \{0°, 18°, \ldots, 130°\}$. This parametrization allows for a total of 36 possible light source combinations. All combinations of two active lights were rendered with four distinct illumination properties, denoted as "global-spec", "global-lamb", "local-spec", and "local-lamb". "global" denotes global illumination via raytracing and includes cast self-shadows. "local" denotes local lighting, i.e., shading is only determined by the surface orientation, and lighting by the angle between incident light $\vec{v}$ and surface normal $\vec{n}$.



(a) Synthetic Face Dataset (ground truth geometry) (b) Extended Yale Face Dataset B

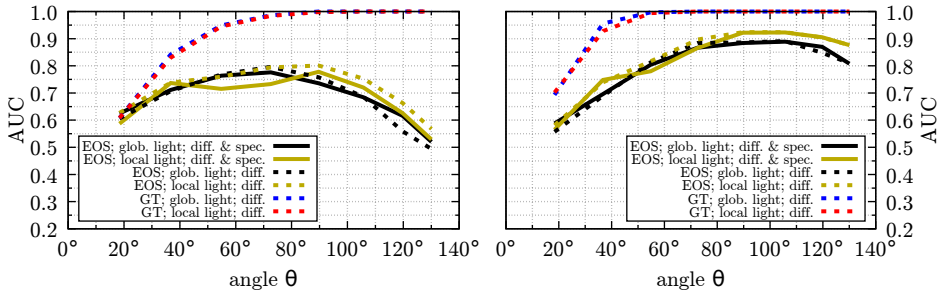
Fig. 4: Detection of inconsistent lighting environments depending on the angular difference between light sources.

"spec" denotes Lambertian and specular reflectance. "lamb" denotes Lambertian reflectance. Additionally, we store the ground truth facial geometry. Only skin pixels are used for further processing, other pixels were ignored. In total, the dataset consists of 432 images. Sample images are shown in Fig. 2 and Fig. 3.

Our experiments on real data are performed on the "Extended Yale Face Database B" [GBK01], consisting of 28 subjects, each illuminated by a light from one out of 64 positions. We use all 1792 frontal view, single light source images. There are in total 2071 possible combinations of light sources. We round θ up to a multiple of 20°. The resulting set of angular distances between lights is $\Theta_{\text{YALE}} = \{0°, 20°, 40°, \ldots, 180°\}$.

6.2 Evaluation Protocol

Evaluation is performed on pairs of randomly chosen (different) subjects, with controlled angular differences between the subjects' lighting environments. Paired lighting environments from the SFD dataset are always rendered with identical options, i.e., "global-spec", "global-lamb", "local-spec", or "local-lamb". Pairs of lighting environments are grouped by their angular distances θ. Each experiment uses N pairs with $\theta = 0°$ and N pairs with $\theta = i \times 20°$ for $i \in \{1, \ldots, 8\}$. For the Extended Yale Face Database B, we use $N_{\text{yale}} = 1700$ and for SFD we chose $N_{\text{SFD}} = 128$. Performances are typically given as area under the curve (AUC). The samples of consistent lighting are identical across experiments. To compute the confidence scores, we use $8 \times N$ different samples of consistent lighting. We set $N_{\text{yale}} = 1500$ and $N_{\text{SFD}} = 64$ due to keep the number of consistent and inconsistent samples balanced.



(a) IBS (b) HIBS $\lambda = 9.0$

Fig. 5: Impact of the reflectance model: local vs. global lighting to model shadows, inclusion or omission of specularities, with estimated ("EOS") or ground truth ("GT") geometry.

6.3 Results

The automated lighting estimation pipeline is set up as follows. Faces are detected using the OpenCV v3.1 library. For each face in the scene, 68 facial landmarks are determined using Dlib [Ki09]. Then, the EOS framework [Hu16] is used to fit the 3-D model. Surface normals and observed intensities are jointly used to estimate the lighting environment (Eqn. 5).

6.3.1 Comparison of Lighting Environment Representations

Figure 4 shows results for comparing the classic spherical harmonics representation (denoted as "SPHEHA") with the intensity binning sphere IBS and the combination of both, HIBS. SPHEHA and HIBS are evaluated once without any regularization (denoted by the regularization weight $\lambda = 0$) and once with a Tikhonov regularizer [JF07a]. In preliminary experiments, we determined that a regularization weight of $\lambda = 9$ worked reasonably well on a range of scenarios, and we continue to use that value.

The results in Fig. 4a are computed on the synthetic dataset with available ground truth geometry. The results in Fig. 4b are computed on the Extended Yale Face Database B. Figure 4a is a best case for all estimators. The AUC is consistently high, and almost perfect for all processing variants at angular differences of about 60°. Figure 4b on real data is the most challenging scenario, which can be seen from the fact that even for lighting environments with an angular distance of 80°, the AUC is still in the range of 0.7.

On SFD, regularized spherical harmonics and HIBS perform very well. On the real data, IBS performs best together with regularized spherical harmonics, but the differences between the approaches are overall less pronounced. The AUC of the spherical harmonics representation is slightly higher, but IBS and HIBS yield quite good results for the confidence score computation below, which is why we believe that both approaches are worth to consider.

$\mathbf{E}\{\kappa_{np}\}$	**IBS**	**SPHEHA** $\lambda = 0$	**SPHEHA** $\lambda = 9$	**HIBS** $\lambda = 0$	**HIBS** $\lambda = 9$
SFD	0.926	0.898	0.954	**0.974**	0.948
ExtYaleB	0.696	0.622	0.687	**0.698**	0.670

Tab. 1: Expected decision confidence score

6.3.2 Error Factorization

Fig. 5 shows a comparison between local and global lighting, diffuse and diffuse+specular reflectance, and estimated geometry ("EOS") versus ground truth geometry ("GT") on the synthetic dataset. The plots consist of two clusters. Using ground truth geometry outperforms all other variants by a large margin. At the same time, the differences between all other variants are minor. Figure 5 shows that accuracy of surface normals has by far the biggest impact on the estimation error, and that limitations in the computational model are of secondary concern.

6.3.3 Confidence Score

Figure 6 shows the confidence scores on the SFD dataset with known geometry and on the Yale dataset. Left, confidence scores for consistent lighting are shown. Right, confidences for inconsistent lighting are shown. On the SFD dataset, HIBS exhibits the steepest transition between consistent and inconsistent lighting. On the Yale dataset, confidence scores are mostly lower. The expected confidence score $E\{\kappa_{np}\}$ incorporates both the confidence score value $\kappa_{np}(\rho)$ and the correlation value density $p(\rho)$,

$$E\{\kappa_{np}\} = \sum_{\rho=-1.0}^{1.0} p(\rho) \cdot \kappa_{np}(\rho) \quad . \tag{6}$$

From a user perspective, higher values $E\{\kappa_{np}\}$ can suggest a more reliable decision. The expectation values in Tab. 1 show that HIBS with $\lambda = 0$ performs best on both datasets.

7 Conclusions

We present three components that support the use of 3-D lighting environments for physics-based detection of image manipulations. First, we propose to include for the similarity computation of lighting environments only angular ranges that are backed up by actual observations (as opposed to extrapolated intensities). Second, we present a method to factorize and analyze the error contributions of the face fitting and correlation computation. It turns out that the impact of violations to the physical model due to specularities and

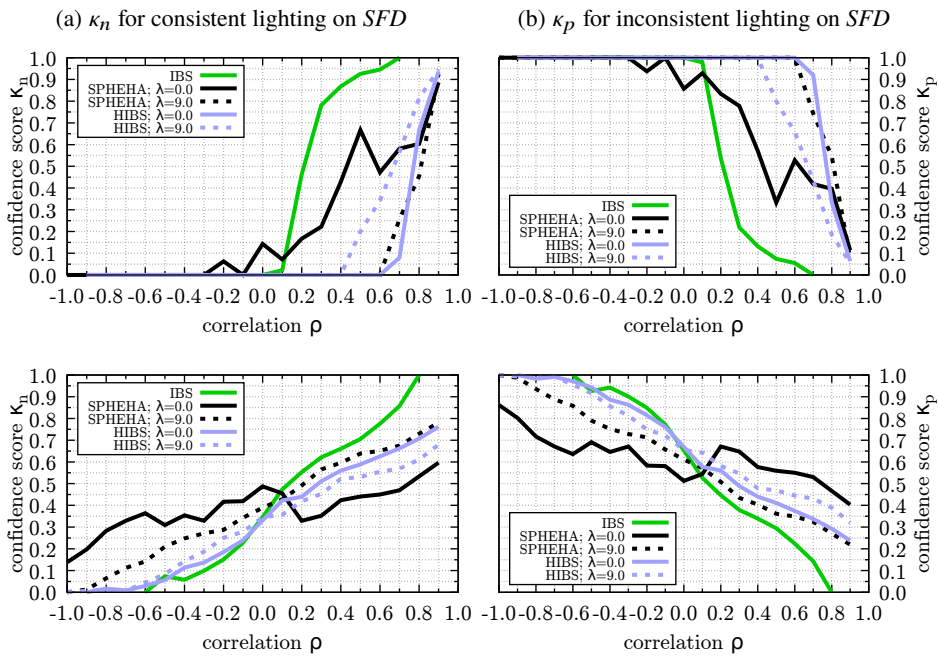


Fig. 6: Confidence scores with known geometry on synthetic data (SFD) and on real-world data (Extended Yale Face Dataset B).

self-shadows is minor compared to geometric fitting errors. This indicates that a high-quality face fit has by far the biggest impact on the overall accuracy. Third, for practical use, we propose a lighting environment confidence score that is learned from the actual data, specifically for the available implementation of the processing pipeline.

Acknowledgements

This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

References

[Fa16] Farid, Hany: Photo Forensics. MIT Press, 2016.

[GBK01] Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):643–660, June 2001.

[Hu16] Huber, P.; Hu, G.; Tena, R.; Mortazavian, P.; Koppen, W.; Christmas, W.; Rätsch, M.; Kittler, J.: A Multiresolution 3D Morphable Face Model and Fitting Framework. In: International Conference on Computer Vision Theory and Applications. pp. 79–86, February 2016.

[JF07a] Johnson, M.; Farid, H.: Exposing Digital Forgeries in Complex Lighting Environments. IEEE Transactions on Information Forensics and Security, 2(3):450–461, September 2007.

[JF07b] Johnson, Micah; Farid, Hany: Exposing Digital Forgeries through Specular Highlights on the Eye. In: Proceedings of the 9th International Workshop on Information Hiding. Saint Malo, France, pp. 311–325, September 2007.

[KF10] Kee, Eric; Farid, Hany: Exposing Digital Forgeries from 3-D Lighting Environments. In: Proceedings of the 2nd IEEE International Workshop on Information Forensics and Security. Seattle, WA, USA, December 2010.

[Ki09] King, Davis E.: Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research, 10:1755–1758, July 2009.

[KOF14] Kee, Eric; O'Brien, James F.; Farid, Hany: Exposing Photo Manipulation from Shading and Shadows. ACM Transactions on Graphics, 33(5):165:1–21, August 2014.

[OF12] O'Brien, James F.; Farid, Hany: Exposing Photo Manipulation with Inconsistent Reflections. ACM Transactions on Graphics, 31(1):1–11, January 2012.

[Pe16] Peng, Bo; Wang, Wei; Dong, Jing; Tan, Tieniu: Automatic Detection of 3-D Lighting Inconsistencies via a Facial Landmark based Morphable Model. In: IEEE International Conference on Image Processing. pp. 3932–3936, September 2016.

[Pe17] Peng, Bo; Wang, Wei; Dong, Jing; Tan, Tieniu: Optimized 3D Lighting Environment Estimation for Image Forgery Detection. IEEE Transactions on Information Forensics and Security, 12(2):479–494, February 2017.

[RA10] Riess, C.; Angelopoulou, E.: Scene Illumination as an Indicator of Image Manipulation. In: Proceedings of the 12th International Conference on Information Hiding. Calgary, AB, Canada, pp. 66–80, June 2010.

[RTD11] Redi, Judith; Taktak, Wiem; Dugelay, Jean-Luc: Digital Image Forensics: A Booklet for Beginners. Multimedia Tools and Applications, 51(1):133–162, January 2011.

[SM13] Sencar, Husrev Taha; Memon, Nasir, eds. Digital Image Forensics. Springer, 2013.

[Zh09] Zhang, Wei; Cao, Xiaochun; Zhang, Jiawan; Zhu, Jigui; Wang, Ping: Detecting Photographic Composites using Shadows. In: Proceedings of the IEEE International Conference on Multimedia and Expo. Cancun, Mexico, pp. 1042–1045, June 2009.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 171

Ich sehe was, das du nicht siehst

Die Realität von Mobilebanking zwischen allgemeinen und rechtlichen Anforderungen

Vincent Haupert[1], Gaston Pugliese[2]

Abstract: Kürzlich hat die Europäische Kommission die Technischen Regulierungsstandards im Rahmen der Zahlungsdiensterichtlinie II vorgelegt. Sie regeln unter anderem auch die Anforderungen an die starke Kundenauthentifizierung, die für digitale Zahlungsvorgänge zumindest eine Zwei-Faktor-Authentifizierung vorschreiben.

Der Beitrag setzt sich mit den rechtlichen Vorgaben auseinander, indem zunächst allgemeine Anforderungen formuliert werden, ehe darauf eingegangen wird, ob und wie Transaktionen auf nur einem mobilen Endgerät diesen Anforderungen genügen können. Hierbei wird die Transaktionssicherheit der Ein-Gerät-Authentifizierung anhand von smsTAN- und App-basierten Mobilebankingverfahren mittels allgemeiner wie auch rechtlicher Anforderungen bewertet. Es zeigt sich, dass die vorherrschenden Plattformen Android und iOS die Anforderung an ein unkopierbares Besitzelement bereits heute erfüllen können, während eine sichere Anzeige weiter eine Zukunftsaufgabe bleibt, gerade auch, weil der Gesetzgeber klare Anforderungen versäumt hat.

Keywords: Mobilebanking; Onlinebanking; Compliance; PSD2; RTS; Trusted Path; Secure Display

1 Einleitung

Die Europäische Bankenaufsicht (EBA) hat im Rahmen ihres Mandats zur Ausarbeitung der Technischen Regulierungsstandards (RTS) der Zahlungsdiensterichtlinie II (PSD2) auf ihr Diskussionspapier einen Rekord von 224 Rückmeldungen zu vermelden [EBA17]. Schon allein die Tatsache, dass ein Regulierungsvorhaben der EBA noch nie so viel Resonanz erzeugt hat, zeigt bereits die Tragweite der im Januar 2018 in Kraft getretenen Richtlinie.

Besonders kontrovers wurden die Anforderungen an die starke Kundenauthentifizierung diskutiert, die auch Einfluss darauf haben, welche Legitimierungsverfahren die Banken zukünftig noch einsetzen dürfen, bzw. welche Anforderungen an neue Verfahren zu stellen sind. Daraus ergeben sich auch unmittelbare Auswirkungen auf die Kunden, da diese unter Umständen nicht nur ihre Legitimierungsverfahren wechseln, sondern auch Verhaltensweisen ändern müssen. Bis vor Kurzem war vor allem unklar, ob die bereits verfügbaren App-basierten Mobilebankinglösungen der Kreditinstitute mit dem Inkrafttreten der RTS noch konform wären, haben die Institute in den vergangenen Jahren doch beträchtliche Ressourcen in deren Entwicklung und Vermarktung investiert.

Lehrstuhl für Informatik 1 (IT-Sicherheitsinfrastrukturen), Department Informatik, Technische Fakultät, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Martensstraße 3, 91058 Erlangen, Deutschland
[1] vincent.haupert@cs.fau.de
[2] gaston.pugliese@cs.fau.de

doi:10.18420/sicherheit2018_13

Mobilebanking meint im Unterschied zum Onlinebanking, dass Transaktionen von ein und demselben mobilen Endgerät durchgeführt werden können. Ein solches Vorgehen stellt im Vergleich zu allen bisherigen Legitimierungsverfahren im Onlinebanking einen Bruch dar, da sie eine physikalische Trennung der Authentifizierungselemente immer als inhärentes Sicherheitsmerkmal verstanden haben. Mit dem Aufkommen von Smartphones haben sich die Voraussetzungen scheinbar geändert, da nahezu jede Bank von Rang und Namen ein Verfahren zum Mobilebanking anbietet.

Obwohl auch die finale Fassung der Europäischen Kommission [COM17] vom 27. November 2017 einigen Interpretationspielraum lässt, kann sie als Entscheidung zugunsten des Status Quo und somit auch der Banken gewertet werden. Der vorliegende Beitrag setzt sich mit den finalen Voraussetzungen an die starke Kundenauthentifizierung auseinander. Hierzu werden zunächst allgemeine Voraussetzungen an die Sicherheit von digitalen Transaktionen formuliert, um im nächsten Abschnitt die Anforderungen der RTS zu skizzieren. An diesen allgemeinen und rechtlichen Anforderungen werden dann SMS- und App-basierte Mobilebankingansätze jeweils gemessen. Den Abschluss bildet ein Ausblick darauf, wie digitale Transaktionen zukünftig über nur ein Smartphone sicher abgewickelt werden können, sowie eine Zusammenfassung unserer Ergebnisse.

2 Allgemeine Anforderungen an die Sicherheit digitaler Transaktionen

In diesem Abschnitt sollen allgemeine Anforderungen formuliert werden, die die Sicherheit digitaler Transaktionen aus Nutzersicht beschreiben. Der Begriff *digitale Transaktion* abstrahiert ganz bewusst von den geläufigen Bezeichnungen *Onlinebanking* und *Mobilebanking* und meint jedwede Aktion, die über einen digitalen Kanal abgewickelt wird und eine Zustandsüberführung herbeiführt. Es sei darauf hingewiesen, dass dies auch Aktionen abseits des prototypischen Beispiels der elektronischen Überweisung einschließt, insofern sie Änderungen an den Kernattributen des Kontos herbeiführen.

Für die Sicherheit von digitalen Transaktionen sind die beiden folgenden Eigenschaften maßgeblich:

1. Eine digitale Transaktion wird nur genau dann durchgeführt, wenn sie vom Nutzer willentlich ausgeführt wurde.
2. Eine vom Nutzer willentlich ausgeführte digitale Transaktion ist manipulationsfrei.

Hierbei fordert 1, dass ein Zugriff auf die am Transaktionsprozess beteiligten Authentifizierungselemente nicht, bzw. nur in solch einer Art und Weise durch unautorisierte Dritte möglich ist, dass keine digitalen Transaktionen durchgeführt werden können. Weiter fordert 2, dass auch dann, wenn ein unautorisierter Dritter keinen direkten Zugriff auf die Authentifizierungselemente im Sinne von 1 erlangen kann, es ihm nicht möglich ist, eine Transaktion für den Nutzer transparent zu manipulieren. Aus dieser Anforderung folgt auch, dass zumindest eines der am digitalen Transaktionsprozess beteiligten Geräte über einen *Trusted Path* verfügt. Hiermit ist ein geschützter, vertrauenswürdiger Kanal gemeint, der

zumindest Authentizität bzgl. der zum Zwecke der Durchführung der digitalen Transaktion ein- und ausgegebenen Daten garantiert, sodass eine sichere Verifikation und Bestätigung durch den Nutzer möglich ist [WW17; Zh12].

3 Rechtliche Anforderungen an die Sicherheit digitaler Transaktionen

Dieser Abschnitt beschäftigt sich mit den Technischen Regulierungsstandards (RTS) der Europäischen Kommission vom 27. November 2017 [COM17], die auf dem Standardentwurf der EBA vom 23. Februar 2017 [EBA17] beruhen. Insofern es binnen drei Monaten seitens des Europäischen Parlaments oder des Rats keine Einwände gibt, tritt die Verordnung mit der Veröffentlichung im Amtsblatt der Europäischen Union in Kraft, gilt jedoch erst 18 Monate später.

3.1 Anforderungen an die Authentifizierungselemente

Die PSD2 fordert ganz allgemein, dass die Authentifizierung zumindest zwei Elemente aus den Kategorien Wissen, Besitz und Inhärenz umfasst. Die Anforderungen an die einzelnen Elemente sind weitgehend intuitiv: So sollten Maßnahmen ergriffen werden, die dem Risiko, dass ein unautorisierter Dritter das Wissenselement in Erfahrung bringt oder das Besitzelement replizieren kann, nach Möglichkeit vollumfänglich entgegenwirken. Ebenso muss es vermieden werden, dass ein als Inhärenz kategorisiertes Authentifizierungselement in unautorisierte Hände gelangt. Im Unterschied zu Wissens- und Besitzelementen ist das Anwenden eines inhärenten Authentifizierungselements regelmäßig mit einem heuristischen Prozess verbunden, der durch eine gewisse Fehlerrate charakterisiert wird. Folgerichtig ist diese Fehlerrate zumindest auf einem sehr niedrigen Niveau zu halten. Ferner sollen die Hard- und Software des entsprechenden Geräts auch sicherstellen, dass ein Inhärenzelement nicht unrechtmäßig verwendet werden kann.

3.2 Authentifizierungscode, dynamische Verknüpfung und Unabhängigkeit

Neben den Bedingungen an die einzelnen Elemente setzen die RTS auch noch einen Authentifizierungscode, eine dynamische Verknüpfung, sowie die Unabhängigkeit der Elemente voraus. Der Authentifizierungscode ist dabei mit einer Transaktionsnummer (TAN), wie sie seit jeher aus dem Onlinebanking bekannt ist, vergleichbar. Im Zusammenspiel mit der dynamischen Verknüpfung muss der Authentifizierungscode zumindest in Abhängigkeit zu Empfänger und Betrag einer Transaktion stehen. Die Anforderung an die dynamische Verknüpfung der Transaktion geht aber noch weiter: Nicht nur müssen dem Zahlungsauslöser zu jeder Phase des Transaktionsprozesses Empfänger und Betrag sichtbar gemacht werden, sondern es müssen auch solche Sicherheitsmaßnahmen ergriffen werden, die dafür Sorge tragen, dass die Vertraulichkeit, Authentizität und Integrität der angezeigten Transaktionsdaten nicht verletzt wird.

Schlussendlich soll die Unabhängigkeit der Elemente sicherstellen, dass die Kompromittierung eines der Authentifizierungselemente nicht auch die Kompromittierung eines oder mehrerer der anderen Elemente nach sich zieht. Zu diesem Zweck sollen Authentifizierungselemente, wann immer sie durch ein Mehrzweckgerät realisiert werden, in getrennten sicheren Ausführungsumgebungen durch Softwaremaßnahmen gekapselt werden. Ferner sollen Maßnahmen ergriffen werden, die sicherstellen, dass keine Software- oder Gerätemodifikationen stattgefunden haben, weder durch unautorisierte Dritte, noch durch den Nutzer selbst. Werden Modifikationen festgestellt, dann sollen die daraus resultierenden Konsequenzen eingedämmt werden.

3.3 Vergleich zu den allgemeinen Anforderungen

Im Unterschied zu unseren allgemeinen Anforderungen, sind die soeben geschilderten rechtlichen Voraussetzungen deutlich spezifischer. Auch fordert die PSD2 explizit eine Mehr-, jedoch mindestens eine Zwei-Faktor-Authentifizierung. Obwohl diese Vorgabe aus Sicht der IT-Sicherheit zweifelsohne sinnvoll ist, folgt sie aus unseren Anforderungen nicht zwangsläufig, insofern beide Bedingungen erfüllt sind.

Dennoch lassen sich die Anforderungen durch die PSD2, bzw. die ihr zugehörigen RTS, jeweils unseren allgemeinen Voraussetzungen an die Sicherheit zuordnen. So entsprechen die Anforderungen an die Authentifizierungselemente, den Authentifizierungscode und die Unabhängigkeit weitgehend unserer Vorgabe, dass die Elemente durch einen unautorisierten Dritten nicht zugreifbar sein dürfen. Die Anforderung an die dynamische Verknüpfung der Transaktion geht sogar noch weiter als unsere Forderung nach Manipulationsfreiheit. Während wir nur einen Trusted Path voraussetzen, damit dem Zahler eine lückenlose Transaktionsverifikation möglich ist, fordert die PSD2 neben Authentizität auch Integrität und Vertraulichkeit. In unserer Anforderung kommt die Integrität nicht vor, da diese vom Nutzer zu prüfen ist. Einzige Voraussetzung ist, dass ein Nutzer auch genau das bestätigt bzw. nicht bestätigt, was ihm dargestellt wird. Obwohl Vertraulichkeit zweifelsfrei ein erstrebenswertes Gut ist, das in der Praxis stets gewahrt werden sollte, spielt sie für die technische Transaktionssicherheit eine untergeordnete Rolle.

4 Bewertung von SMS- und App-basierten Mobilebankingverfahren

Nachdem die allgemeinen sowie rechtlichen Anforderungen geklärt sind, setzen wir uns im Folgenden damit auseinander, ob diese in der Praxis des Mobilebankings eingehalten werden können. Wie eingangs beschrieben, sind mit Mobilebanking Transaktionen gemeint, die auf nur einem Gerät stattfinden. In der Realität moderner Smartphones qualifizieren sich unter den gebräuchlichen Legitimierungsverfahren nur das smsTAN- sowie App-basierte Verfahren, die laut einer repräsentativen Umfrage aus dem Jahre 2016 von 36% bzw. $5-8\%$[3]

[3] Die Spannweite resultiert aus der separat erfassten Verwendung des photoTAN-Verfahrens. Das photoTAN-Verfahren wird sowohl als dediziertes Gerät als auch als App-basiertes Verfahren angeboten. Die Studie gibt jedoch keinen genaueren Aufschluss darüber, welche Anteile auf welche Ausprägung entfallen. Die Hardwarevariante wäre ihren Eigenschaften nach eher dem chipTAN-Verfahren zuzuordnen.

der Bevölkerung verwendet werden [NB16]. Der Rest entfällt auf das chipTAN- bzw. iTAN-Verfahren, die beide jedoch nicht durch ein einziges Gerät abgebildet werden.

4.1 Angreifermodell

Obwohl gerade die allgemeinen Anforderungen hinreichende und notwendige Bedingungen für die theoretische Sicherheit von Transaktionen sind, können sie in der Praxis der Transaktionssicherheit digitaler Bankgeschäfte nur durch ein entsprechendes Angreifermodell bewertet werden. Die Schwierigkeit besteht darin, ein solches Angreifermodell zu wählen, das die in der Wirklichkeit herrschenden Umstände möglichst gut widerspiegelt [HHF17]. Zu diesem Zweck stellen wir die folgenden Bedingungen an das Angreifermodell, die unseren formalen Anforderungen gegenübergestellt werden:

1. Ein Angreifer hat keinen physischen Zugriff auf die Authentifizierungselemente.
2. Der Nutzer kommt seiner Sorgfaltspflicht nach, sodass der digitale Transaktionsprozess von seiner Seite regelkonform erfolgt.
3. Der Angriff enthält zumindest eine technische Komponente, basiert also nicht ausschließlich auf der Täuschung des Nutzers im Sinne von *Social Engineering*.

Im Folgenden wird ferner davon ausgegangen, dass die Transaktionsauslösung und -bestätigung auf ein und demselben Smartphone stattfindet. Hierbei wird ein System der beiden vorherrschenden Plattformen Android und iOS angenommen. Die Transaktionsauslösung kann dabei entweder aus dem Browser oder einer eigenen Banking-App erfolgen. Während dieser Vorgang für das smsTAN-Verfahren zwingend ist, muss die Bestätigung der Transaktion nicht zwangsläufig durch eine separate App erfolgen, sondern kann auch von derselben App durchgeführt werden, die bereits zur Transaktionsauslösung verwendet wurde.

4.2 Die smsTAN und die Geräteevolution

Beim smsTAN-Verfahren wird dem Nutzer eine TAN samt Empfänger und Betrag über den Mobilfunk zugestellt, die dann nach erfolgreicher Verifikation der Transaktionsdetails manuell in den transaktionsauslösenden Kanal zu übertragen ist. Als das smsTAN-Verfahren 2003 eingeführt wurde [DP03], waren die die SMS-empfangenden Geräte noch merklich in ihrer Funktionalität eingeschränkt, weshalb sie heute in der Vorherrschaft der Smartphones als Featurephones bezeichnet werden. Ihre primären Aufgaben waren das Telefonieren und das Senden und Empfangen von SMS. Beide Aufgabenfelder sind auf modernen Smartphones deutlich in den Hintergrund gerückt. Dennoch existiert die gleiche Funktionalität auch heute noch auf den mobilen Endgeräten und wird auch nach wie vor von einer SIM-Karte abgewickelt. Hierbei hat die sie tragende Hardware eine Veränderung weg von den Spezialgeräten der Featurephones, hin zu den Mehrzweckgeräten moderner Smartphones

vollzogen. Zu den wesentlichen Neuerungen von Smartphones zählen die quasi immer verfügbare Internetverbindung sowie ein Ökosystem aus verschiedenen Apps, die eine bestimmte Funktionalität bereitstellen.

Bei den Featurephones sorgte die fehlende Internetfähigkeit unter der Verwendung des smsTAN-Verfahrens noch implizit dafür, dass zwei unterschiedliche Geräte an der Transaktionsdurchführung beteiligt waren. Mit dem Aufkommen von Smartphones wurde diese implizite Gerätetrennung aufgehoben, weil es jetzt theoretisch möglich war, Transaktionen von ein und demselben Gerät durchzuführen. Es ist unmittelbar begreifbar, dass dieser Zustand zu einer Absenkung des Sicherheitsniveaus führte und man sich jetzt sogar wieder einem Bedrohungsszenario gegenübersah, das es mit der Einführung des smsTAN-Verfahrens gerade zu bekämpfen galt: nämlich die Infektion mit Schadsoftware [CT09]. Im Vergleich zum iTAN-Verfahren hatte sich die Situation sogar noch verschlechtert: Während die reine Infektion des Nutzercomputers nur dazu genutzt werden konnte, Transaktionen transparent zu manipulieren, war durch die Verwendung des smsTAN-Verfahrens auf einem Smartphone sogar das eigenständige Ausführen beliebiger Transaktionen durch entsprechende Schadsoftware zu beliebigen Zeitpunkten denkbar.

Dieser Umstand ist auch der Deutschen Kreditwirtschaft (DK) nicht verborgen geblieben. Die DK ist ein Interessensverbund eines Großteils der öffentlich-rechtlichen, genossenschaftlich und privat organisierten Banken. Auch die DK, damals noch als Zentraler Kreditausschuss bekannt, wollte dieser neuen Entwicklung Rechnung tragen und schrieb schon 2008 für die Verwendung des smsTAN-Verfahrens zwei Geräte vor [ZKA08]. Dies begründete sie damit, dass die Sicherheit von Transaktionen maßgeblich davon abhängt, dass „man sich der Technik der Übertragung über zwei unterschiedliche Kanäle“ bedient. Demnach müsse Mobilebanking mit dem smsTAN-Verfahren „in den Kundenbedingungen für das Online-Banking explizit ausgeschlossen“ werden [DK].

Diese Vorgabe führt dazu, dass den Kunden, die sich mit ihrem Smartphone über den Browser oder die Banking-App des Kreditinstituts anmelden, die Möglichkeit, Überweisungen zu tätigen, ausgeblendet wird, insofern sie das smsTAN-Verfahren bei einer Bank verwenden, die direkt oder indirekt in der DK organisiert ist. Obwohl diese Regelung zunächst sinnvoll und konsequent erscheint, stellt sich durch sie kein Sicherheitsgewinn, sondern nur ein Verlust an Benutzerfreundlichkeit ein. Dem Nutzer wird nämlich nicht untersagt, beispielsweise seinen Kontostand über den mobilen Browser oder die Banking-App zu überprüfen. Um Zugang zu diesem zu erhalten, muss der Kunde jedoch seine Zugangsdaten eingeben. Sollte das Handy durch eine entsprechende Schadsoftware infiziert sein, kann diese die Zugangsdaten mitschneiden. Der Angreifer unterliegt jedoch nicht den gleichen Restriktionen wie der Nutzer: Oft erkennt der Bankenserver anhand des User-Agents des Browsers oder durch ein bestimmtes Protokoll der Banking-App, dass sich der Nutzer über ein mobiles Endgerät Zugang zu seinem Konto verschafft. Eine Schadsoftware könnte nun ohne Weiteres z. B. den User-Agent seiner HTTP(S)-Anfragen so aussehen lassen, dass der Bankenserver den Nutzer an einem Desktop-Computer vermutet. In Folge wäre die Vorgabe, zwei Geräte zu verwenden, erfüllt und eine Transaktionsauslösung mit Bestätigung über das smsTAN-Verfahren wird möglich. Da sich die Schadsoftware aber in Wirklichkeit auf

dem Smartphone befindet, kommt die via SMS zugestellte TAN dennoch auf dem gleichen Endgerät an. Ein Angreifer kann somit beliebige Transaktionen tätigen, wenn es ihm gelingt, dass Gerät zu kompromittieren.

Gerade die rasant steigende Komplexität der ersten Smartphones sorgte auch für die ersten Schadensfälle [Dm14]. Die SIM-Karte, die letztendlich dafür verantwortlich ist, die SMS entgegenzunehmen, blieb uns jedoch bis heute erhalten und hat somit die Geräteentwicklung vom Spezial- zum Mehrzweckgerät ohne merkliche Änderungen auf Seiten des Geräts mitgemacht. Es ist zwar richtig, dass es in der Vergangenheit immer wieder erfolgreiche Angriffe auf die technische Infrastruktur der SMS gegeben hat [Mu13; Re16; RKH16; SZ17b] und dass sogar neuere Mobilfunkstandards nicht nur alte Sicherheitsprobleme schließen, sondern auch neue schaffen [Tu16]. Dennoch wurde die Sicherheit des smsTAN-Verfahrens gerade durch den Betrieb auf einem Mehrzweckgerät erodiert.

4.3 App-basierte Verfahren

Bei den App-basierten Methoden haben sich verschiedene Verfahren herausgebildet. Allen gemein ist jedoch, dass die für die Transaktionsverifikation benötigten Daten nicht direkt über den Mobilfunk, sondern über ein IP-basiertes Protokoll übertragen werden. Im Gegensatz zum smsTAN-Verfahren ist hierfür nicht zwangsläufig eine SIM-Karte notwendig, obwohl auch eine Abwicklung über das Internetangebot des Mobilfunkanbieters möglich ist. Für das App-basierte Verfahren ist dies jedoch transparent. Der Unterschied der einzelnen App-basierten Verfahren liegt in ihrer Ausprägung (eine oder zwei Apps) und ihres Rückkanals (online oder offline bzw. manuell). Obwohl die seit jeher herrschende Zwei-Faktor-Authentifizierung im Onlinebanking stets eine TAN beinhaltet hat, erwächst aus der Zahlenfolge per se kein Sicherheitsgewinn, da sie semantisch wertlos ist. Sie ist vielmehr ein Artefakt derer Verfahren, die keinen digitalen Rückkanal bieten. Aus diesem Grund fordern App-basierte Verfahren zunehmend nicht mehr, dass die TAN manuell in den transaktionsauslösenden Kanal übertragen wird, bzw. verzichten komplett auf eine für den Nutzer sichtbare TAN.

Der Blick in die Finanzbranche lässt erkennen, dass großes, in jedem Fall aber größeres, Vertrauen in die Sicherheit von Smartphones herrscht, als das bei stationäre Computer und Notebooks der Fall ist. Bis vor Kurzem wäre es noch undenkbar gewesen, alle am Transaktionsprozess beteiligten Authentifizierungselemente über ein und dasselbe Gerät abzubilden, ohne dass dem ein entscheidender Sprung in Sachen Hardware- und Softwaresicherheit vorangegangen wäre, der diesem einschneidenden Schritt Rechnung getragen hätte. Nichtsdestotrotz sind die hierbei entstandenen und entstehenden Verfahren mittlerweile allesamt so ausgelegt, dass sie auch für das Nutzungsszenario des Mobilebankings geeignet sind. Ruft man sich den vorhergehenden Abschnitt und die Aussage der DK bzgl. der Verwendung des smsTAN-Verfahrens auf nur einem Gerät in Erinnerung, überrascht dieses Vorgehen.

Ein genauerer Blick auf das smsTAN-Verfahren offenbart heute große Ähnlichkeiten zu App-basierten Verfahren, besonders zu solchen, die über zwei eigenständige Apps realisiert

sind. Dies ist darauf zurückzuführen, dass die Funktionalität, die die SMS darstellt, ebenfalls über eine App erfolgt. Die Unterschiede liegen vor allem bei den fehlenden Garantien bzgl. des Zustellwegs und bei der Tatsache, dass die SMS-empfangende App nicht von der zugehörigen Bank stammt. Obwohl es richtig ist, dass die App-basierten Verfahren der Kreditinstitute den Schutzzielen der Authentizität, Vertraulichkeit und Integrität besser Rechnung tragen, bleibt die für das smsTAN-Verfahren aufgestellte Argumentation der DK, dass Transaktionsauslösung und -bestätigung nicht auf einem Gerät erfolgen dürfen, gültig.

4.4 Bewertung

Im Kern lassen sich für sicheres Mobilebanking auf dem Smartphone in Bezugnahme auf unsere allgemeinen und die rechtlichen Anforderungen an die Transaktionssicherheit die folgenden notwendigen Voraussetzungen formulieren:

1. Das Besitzelement lässt sich unter praktischen Gesichtspunkten nicht replizieren, sodass eine autonome Inbetriebnahme ausgeschlossen ist.
2. Das Smartphone garantiert Verifizierbarkeit der Transaktionsdetails, sodass die dem Nutzer dargestellten Daten auch denen entsprechen, die der Bank vorliegen, sollte der Nutzer die Transaktion freigeben.

Nachdem unsere allgemeinen Voraussetzungen keine Vertraulichkeit vorschreiben, die Sicherheit eines Wissenselements, wie es im heutigen Mobilebanking üblich ist, jedoch ausschließlich auf Vertraulichkeit fußt, werden wissensbasierte Authentifizierungselemente als kompromittiert betrachtet. Da die RTS hier weiter gehen, unser Angreifermodell jedoch auch nicht-technische Angriffe erlaubt, insofern diese nicht ausschließlich vorkommen, kann auch im Rahmen der rechtlichen Voraussetzungen die Annahme getroffen werden, dass das Wissenselement – in der Regel Benutzername und Passwort – nicht mehr nur dem autorisierten Nutzer bekannt ist.

Im Bezug auf die in 1 geforderte Nichtkopierbarkeit des Besitzelements ergeben sich für die SMS- und App-basierten Verfahren unterschiedliche Situationen. Das smsTAN-Verfahren erfüllt diese Voraussetzung, weil die SIM-Karte letztendlich ein Mikroprozessor ist, dessen Replikation aus der Ferne ohne physikalischen Zugriff ausgeschlossen ist. Es ist zwar richtig, dass es in der Vergangenheit schon Situationen gegeben hat, in denen sich Kriminelle Zugriff auf eine Ersatzkarte des Nutzers verschaffen konnten, dies erfolgte jedoch fernab des Verantwortungsbereichs des Nutzers [SZ17a]. Die Nichtkopierbarkeit im Sinne von 1 bleibt hiervon unberührt.

Für App-basierte Verfahren kann keine derart pauschale Aussage getroffen werden, da die theoretischen Möglichkeiten zur Aufrechterhaltung von praktischer Nichtkopierbarkeit auf den aktuell eingesetzten Smartphones zwischen unmöglich, mäßig und vollständig reichen. Grundsätzlich ist die Ausgangssituation fundamental anders als bei einer SIM-Karte, welche bereits ab ihrer Entgegennahme personalisiert ist. Die App-basierten Legitimierungsverfahren müssen aber zuerst über einen Einrichtungsprozess personalisiert werden, da alle

Apps aus den offiziellen Bezugsquellen zunächst gleich sind. Die im Rahmen des Einrichtungsprozesses hinterlegten Daten werden oft lediglich in einem privaten, App-eigenen Verzeichnis abgelegt. Dieses Vorgehen zieht eine triviale Eins-zu-Eins-Kopierbarkeit nach sich: Werden die nach der Installation angelegten Daten kopiert und auf einem anderen Gerät wiederhergestellt, ergibt sich der gleiche Zustand auf dem Ziel- wie auf dem Quellgerät. Daraus folgt auch, dass die App auf dem Zielgerät in einem registrierten Zustand verwendet werden kann. Diese Situation versuchen die herausgebenden Banken freilich zu vermeiden, indem sie eine Gerätebindung einführen.

Um eine Gerätebindung herzustellen, kann die App zum Beispiel auf *Fingerprinting* zurückgreifen. Zu diesem Zweck liest die App während des ersten Starts – oder auch während des Registrierungsprozesses – verschiedene Umgebungswerte aus, die dann auch in den privaten App-Daten gespeichert werden. Diese Werte werden nun periodisch, z. B. beim Start der App, erneut abgerufen und mit den Werten verglichen, die zuvor gespeichert wurden. Weichen die Werte ab, schließt der Algorithmus zur Umsetzung der Gerätebindung unter Umständen darauf, dass die App-Instanz kopiert wurde. Die Schwierigkeit an diesem Vorgehen liegt daran, möglichst viele Werte zu verwenden, die sowohl einzigartig als auch stabil sind. Es hat sich gezeigt, dass die Hersteller hier einen eher defensiven Ansatz bevorzugen und nur wenige, dafür aber stabile und einzigartige, Werte verwenden [HM18]. Das führt aber auch dazu, dass sie durch statische und dynamische Analyse relativ leicht bestimmt werden können, um so die Gerätebindung zu umgehen.

Eine adäquate Variante Gerätebindung herzustellen, ist die Verwendung von speziellen Hardwarebausteinen in aktuellen Smartphones, mit denen asymmetrische Kryptographie möglich ist. Wichtig ist, dass die so erstellten privaten Schlüssel den gesicherten Bereich niemals verlassen. Auf diese Art kann selbst dann der private Schlüssel nicht extrahiert werden, wenn das komplette Betriebssystem bis hin zum Kernel-Level kompromittiert ist. Eine stabile Verfügbarkeit von solchen Verfahren ist seit Android 6 (hardwaregestützter KeyStore) und seit iOS 9 (KeyChain mit Secure Enclave) gegeben. Um nun Gerätebindung unter solchen Umständen zu erreichen, müssen die am Authentifizierungsprozess als Besitzelemente beteiligten Apps im Registrierungsprozess ein asymmetrisches Schlüsselpaar generieren und den öffentlichen Schlüssel an die Stelle weitergeben, die die Antworten der App letztendlich bearbeitet. Von diesem Punkt an versieht das Besitzelement alle ausgebenden Daten mit einer Signatur, die von der entgegennehmenden Stelle – z. B. dem Bankserver – stets zuerst geprüft wird. Durch dieses Vorgehen werden zwar keine Aussagen über die gesendeten Daten gemacht, es kann aber zumindest sichergestellt werden, dass eine Nachricht von einem bestimmten Gerät stammt. Es zeigt sich also, dass sich Punkt 1 – unter bestimmten Voraussetzungen – auch auf aktuellen Smartphones unter Verwendung App-basierter Verfahren erfüllen lässt. Die Voraussetzungen unter Android erfüllen zum 8. Februar 2018 bereits 57,7% [An], während es unter iOS sogar mindestens 93% sind [Ap].

Eine ganz andere Situation stellt sich aber für Anforderung 2 ein, die eine sichere Anzeige in dem Sinne fordert, dass der Nutzer genau das sieht, was er gegenüber der Bank dann auch bestätigt. Für die smsTAN stellt schon der Transportweg eine Herausforderung dar, da zumindest den vermittelnden Stellen der Klartext vorliegt und auch keine Garantien bzgl.

der Authentizität gewährleistet werden [Re16]. Es wäre also denkbar, dass der Inhalt der SMS bereits vor Zustellung auf dem Smartphone manipuliert wurde. Bei den App-basierten Verfahren kann die Authentizität relativ einfach hergestellt werden, indem z. B. auf TLS zurückgegriffen wird.

Ist die SMS jedoch einmal auf dem Gerät eingegangen, unterscheiden sich die weiteren Voraussetzungen im Vergleich zu App-basierten Verfahren nicht mehr. Als nächstes müssen dem Nutzer die Transaktionsdetails dargestellt werden, damit er deren Richtigkeit überprüfen kann. Hierfür ist es zwingend notwendig, dass die dargestellten Daten zum einen dem entsprechen, was empfangen wurde, und zum anderen auch nicht mehr verändert werden können, wenn die Transaktionsverifikation vom Nutzer abgeschlossen wurde. Es wurde in der Vergangenheit mehrmals gezeigt, dass solche Garantien insbesondere unter Android aktuell nicht existieren. So ist es einem Angreifer aktuell möglich, die ein- und ausgegebenen Daten zu modifizieren, wenn ein Angreifer Kontrolle über die App oder das Betriebssystem erlangt [Ha17; HM16; HM18]. Unter Android haben Fratantonio et al. erst kürzlich Wege gefunden, um auch innerhalb des vorgeschriebenen Rechtemodells volle Kontrolle über die Benutzerschnittstelle zu erlangen [Fr17].

5 Zusammenfassung und Ausblick

In diesem Beitrag haben wir allgemeine Anforderungen an die Sicherheit von digitalen Transaktionen formuliert und diese den rechtlichen Voraussetzungen im Rahmen der Zahlungsdiensterichtlinie II (PSD2) gegenübergestellt. Dabei zeigte sich, dass sich die rechtlichen Anforderungen im Kern mit unseren allgemeinen Voraussetzungen vereinen lassen. Hierbei wurden mit dem Blick auf das Mobilebanking insbesondere die Nichtkopierbarkeit und die sichere Anzeige als notwendige Bedingungen für sichere Transaktionen auf nur einem mobilen Endgerät ermittelt. Die Bewertung ergab, dass sich die Anforderung an die Nichtkopierbarkeit bereits jetzt gut realisieren lässt. Die Voraussetzung einer sicheren Anzeige, die verhindert, dass ein Nutzer andere Daten bestätigt, als ihm angezeigt werden, bleibt jedoch ein ungelöstes Problem.

Und obwohl wir die Anforderung der Technischen Regulierungsstandards (RTS) bzgl. der dynamischen Verknüpfung so auffassen, dass eine sichere Anzeige gewährleistet werden muss, scheint der Entwicklungsverlauf der RTS eher darauf hinzudeuten, dass es sich hierbei doch nicht um eine strikte Vorgabe handelt. So zeigt der finale Entwurf nicht nur, dass nun keine separaten vertrauenswürdigen Ausführungsumgebungen (TEE) gefordert werden, sondern die EBA stellt auch klar, dass sie es nicht für nötig hält, dass die verschiedenen Authentifizierungselemente über unterschiedliche Geräte abgebildet werden [EBA17].

Wir sind der Auffassung, dass eine Forderung nach einer TEE, wie sie bereits heute in allen Smartphones vorhanden ist, die Entwicklung einer gemeinsamen Betriebssystemschnittstelle signifikant beschleunigt hätte. Stattdessen wird auch in Zukunft Schadsoftware noch auf absehbare Zeit Verstecken spielen können.

Literatur

[An] Android Developers: Dashboards: Platform versions, URL: https://developer.android.com/about/dashboards/index.html, Stand: 08. 02. 2018.

[Ap] Apple Inc.: App Store - Support - Apple Developer, URL: https://developer.apple.com/support/app-store/, Stand: 08. 02. 2018.

[COM17] Europäische Kommission: Delegierte Verordnung (EU) der Kommission zur Ergänzung der Richtlinie (EU) 2015/2366 des Europäischen Parlaments und des Rates durch technische Regulierungsstandards für eine starke Kundenauthentifizierung und für sichere offene Standards für die Kommunikation, Nov. 2017, URL: https://ec.europa.eu/transparency/regdoc/rep/3/2017/DE/C-2017-7782-F1-DE-MAIN-PART-1.PDF.

[CT09] Bachfeld, D.: Mit guten Karten: Sicher bezahlen im Internet. c't 19/26, S. 92–95, 2009.

[DK] Die Deutsche Kreditwirtschaft: mobileTAN, URL: https://die-dk.de/zahlungsverkehr/electronic-banking/mobiletan, Stand: 14. 12. 2017.

[Dm14] Dmitrienko, A.; Liebchen, C.; Rossow, C.; Sadeghi, A.: On the (In)Security of Mobile Two-Factor Authentication. In: Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers. S. 365–383, 2014, URL: https://doi.org/10.1007/978-3-662-45472-5_24.

[DP03] Deutsche Postbank AG: Online Durchstarten mit Postbank direkt: Neues Finanzportal im Internet / Premiere für die mobileTAN in Deutschland, Nov. 2003, URL: https://www.postbank.de/postbank/pr_presseinformation_2003_11284.html, Stand: 14. 12. 2017.

[EBA17] European Banking Authority: Final Report Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2), Feb. 2017, URL: https://www.eba.europa.eu/documents/10180/1761863/Final+draft+RTS+on+SCA+and+CSC+under+PSD2+%28EBA-RTS-2017-02%29.pdf.

[Fr17] Fratantonio, Y.; Qian, C.; Chung, S. P.; Lee, W.: Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. S. 1041–1057, 2017, URL: https://doi.org/10.1109/SP.2017.39.

[Ha17] Haupert, V.: Die fabelhafte Welt des Mobilebankings, 34th Chaos Communication Congress, 34c3: tuwat, Leipzig, Germany, December 27-30, 2017, Chaos Computer Club e.V., Dez. 2017, URL: https://doi.org/10.5446/34946.

[HHF17] Hoffmann, J.; Haupert, V.; Freiling, F.: Anscheinsbeweis und Kundenhaftung beim Online-Banking. Zeitschrift für das gesamte Handels- und Wirtschaftsrecht (ZHR) 181/5, S. 780–816, 2017.

[HM16] Haupert, V.; Müller, T.: Auf dem Weg verTAN: Über die Sicherheit App-basierter TAN-Verfahren. In: Sicherheit 2016: Sicherheit, Schutz und Zuverlässigkeit, Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 5.-7. April 2016, Bonn. S. 101–112, 2016.

[HM18] Haupert, V.; Müller, T.: On App-based Matrix Code Authentication in Online Banking. In (Furnell, S.; Mori, P.; Camp, O., Hrsg.): Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira, Portugal, February 22-24, 2018. S. 149–160, 2018, ISBN: 978-989-758-282-0, URL: https://doi.org/10.5220/0006650501490160.

[Mu13] Mulliner, C.; Borgaonkar, R.; Stewin, P.; Seifert, J.: SMS-Based One-Time Passwords: Attacks and Defense - (Short Paper). In: Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA 2013, Berlin, Germany, July 18-19, 2013. Proceedings. S. 150–159, 2013, URL: https://doi.org/10.1007/978-3-642-39235-1_9.

[NB16] norisbank GmbH: norisbank-Umfrage zum Thema Online-Banking, Nov. 2016, URL: https://www.norisbank.de/ueberuns/presseinformation-norisbank-umfrage-online-banking-ein-viertel-der-deutschen-nutzt-veraltetes-tan-verfahren.html, Stand: 13. 12. 2017.

[Re16] Reaves, B.; Scaife, N.; Tian, D.; Blue, L.; Traynor, P.; Butler, K. R. B.: Sending Out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In: IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016. S. 339–356, 2016, URL: https://doi.org/10.1109/SP.2016.28.

[RKH16] Rao, S. P.; Kotte, B. T.; Holtmanns, S.: Privacy in LTE networks. In: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, MobiMedia 2016, Xi'an, China, June 18-20, 2016. S. 176–183, 2016, URL: http://dl.acm.org/citation.cfm?id=3021417.

[SZ17a] Freiberger, H.; Martin-Jung, H.: Die Tücken der TAN. Süddeutsche Zeitung 73/102, Mai 2017.

[SZ17b] Tanriverdi, H.; Zydra, M.: SMS von gestern Nacht. Süddeutsche Zeitung 73/101, Mai 2017.

[Tu16] Tu, G.; Li, C.; Peng, C.; Li, Y.; Lu, S.: New Security Threats Caused by IMS-based SMS Service in 4G LTE Networks. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. S. 1118–1130, 2016, URL: http://doi.acm.org/10.1145/2976749.2978393.

[WW17] Weiser, S.; Werner, M.: SGXIO: Generic Trusted I/O Path for Intel SGX. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY 2017, Scottsdale, AZ, USA, March 22-24, 2017. S. 261–268, 2017, URL: http://doi.acm.org/10.1145/3029806.3029822.

[Zh12] Zhou, Z.; Gligor, V. D.; Newsome, J.; McCune, J. M.: Building Verifiable Trusted Path on Commodity x86 Computers. In: IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA. S. 616–630, 2012, URL: https://doi.org/10.1109/SP.2012.42.

[ZKA08] Zentraler Kreditausschauss: Mindestsicherheitsanforderungen an die mobile TAN, Apr. 2008, URL: https://die-dk.de/media/files/Mindestsicherheitsanforderungen_mobileTAN_V1_20110621.pdf.

Danksagungen: Wir danken Prof. Dr.-Ing. Felix Freiling für hilfreiche Diskussionen sowie den zahlreichen Gutachtern, die ausführliche und wertvolle Verbesserungsvorschläge geliefert haben. Diese Arbeit entstand im Rahmen des von der DATEV eG geförderten Forschungsprojekts „Softwarebasierte Härtungsmaßnahmen für mobile Applikationen“.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 183

Auf dem Weg zu sicheren abgeleiteten Identitäten mithilfe der Payment Service Directive 2

Daniel Träder[1], Alexander Zeier[2], Andreas Heinemann[3]

Abstract: Online-Dienste erfordern eine eindeutige Identifizierung der Benutzer und somit eine sichere Authentisierung. Insbesondere eGovernment-Dienste innerhalb der EU erfordern eine starke Absicherung der Benutzeridentität. Auch die mobile Nutzung solcher Dienste wird bevorzugt. Das Smartphone kann hier als einer der Faktoren für eine Zwei-Faktor-Authentifizierung dienen, um eine höhere Sicherheit zu erreichen. Diese Arbeit schlägt vor, den Zugang und die Nutzung einer abgeleiteten Identität mit einem Smartphone zu sichern, um es dem Benutzer zu ermöglichen, sich auf sichere Weise gegenüber einem Online-Dienst zu identifizieren. Dazu beschreiben wir ein Schema zur Ableitung der Identität eines Benutzers mithilfe eines Account Servicing Payment Service Provider (ASPSP) unter Verwendung der Payment Service Directive 2 (PSD2) der Europäischen Union. PSD2 erfordert eine Schnittstelle für Dritte, die von ASPSPs implementiert werden muss. Diese Schnittstelle wird genutzt, um auf die beim ASPSP gespeicherten Kontoinformationen zuzugreifen und daraus die Identität des Kontoinhabers abzuleiten. Zur Sicherung der abgeleiteten Identität ist der Einsatz von FIDO (Fast Identity Online) vorgesehen. Wir bewerten unseren Vorschlag anhand der Richtlinien von *eIDAS LoA* (Level of Assurance) und zeigen, dass für die meisten Bereiche das Vertrauensniveau *substantiell* erreicht werden kann. Um diesem Level vollständig gerecht zu werden, ist zusätzlicher Arbeitsaufwand erforderlich: Zunächst ist es erforderlich, Extended Validation-Zertifikate für alle Institutionen zu verwenden. Zweitens muss der ASPSP sichere TAN-Methoden verwenden. Schließlich kann der Widerruf einer abgeleiteten Identität nicht erfolgen, wenn der Benutzer keinen Zugriff auf sein Smartphone hat, das mit der abgeleiteten ID verknüpft ist. Daher ist ein anderes Widerrufsverfahren erforderlich (z. B. eine Support-Hotline).

Keywords: Abgeleitete Identitäten; PSD2; eIDAS; eGovernment; Identitätsmanagement

1 Einleitung

Immer mehr Behörden bieten aufgrund der fortschreitenden Digitalisierung eGoverment-Dienste an. Bei der Online-Authentifizierung werden hierbei oft noch Benutzername und Passwort verwendet, obwohl diese Methode zunehmend als unsicher angesehen wird. Alternative Technologien, wie die Online-Authentifizierung des deutschen Personalausweises

[1] Hochschule Darmstadt, Fachbereich Informatik, Haardtring 100, 64295 Darmstadt, Deutschland daniel.traeder@h-da.de
[2] Hochschule Darmstadt, Fachbereich Informatik, Haardtring 100, 64295 Darmstadt, Deutschland alexander.zeier@h-da.de
[3] Hochschule Darmstadt, Fachbereich Informatik, Haardtring 100, 64295 Darmstadt, Deutschland andreas.heinemann@h-da.de

doi:10.18420/sicherheit2018_14

(nPA), sind noch nicht weit verbreitet. Solche Technologien geben dem Benutzer die Möglichkeit, sich durch Online-Dienste mit hoher Sicherheit identifizieren zu lassen, werden aber derzeit von den deutschen Bürgern nur wenig genutzt (vgl. [Va15]). Einer der Gründe ist die mangelnde Benutzerfreundlichkeit, die durch das gegebene Gesamtsystem verursacht wird (vgl. [WHM16]).

Durch die hohe Verbreitung von Smartphones (vgl. [rB16, Po16]) wird die mobile Benutzerauthentifizierung häufig favorisiert. Außerdem kann das Smartphone als einer der Faktoren (Besitz) in einer Zwei-Faktor-Authentifizierung verwendet werden, was zu einer höheren Sicherheit führt. In Verbindung mit einer abgeleiteten Identität[4] ermöglicht ein Smartphone dem Benutzer, sich gegenüber einem Online-Dienst zu identifizieren.

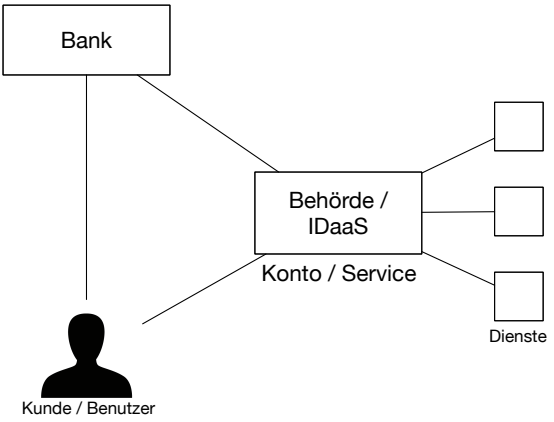


Abb. 1: Szenario zur Ableitung einer Identität bei einem ASPSP

Wir präsentieren ein Schema zur Ableitung einer Identität durch eine Schnittstelle der Payment Service Directive 2 (PSD2), die Kontoinformationen bereitstellt. Diese Informationen können von einem Identity Provider (IdP) gespeichert und für eine spätere Verwendung abgerufen werden. Dieses Szenario ist in Abbildung 1 dargestellt. Eine Behörde oder auch ein Unternehmen möchte die Identität eines (potentiellen) Benutzers in Erfahrung bringen. Der Benutzer ist ebenso Kunde einer Bank. Die Behörde / das Unternehmen nutzt die Bankdaten des Benutzers um diesen zu Authentifizieren. Diese Bankdaten können als Identität in ein Konto gespeichert werden, um einen SSO einzurichten. Dienste müssen in der Lage sein, diesen Identitäten zu vertrauen. Zu diesem Zweck muss ein angemessenes Maß an Vertrauen in die Identität vorhanden sein, welches in der vorliegenden Arbeit bewertet wird. Ein weiterer denkbarer Anwendungsfall wäre, dass ein Unternehmen anderen Unternehmen *Identity-as-a-Service* anbietet, ohne dass Benutzerdaten in einem Konto zwischengespeichert werden.

[4] *A credential issued based on proof of possession and control of a token associated with a previously issued credential, so as not to duplicate the identity proofing process*; siehe hierzu [Bu11]

Für die Evaluierung unseres Schemas konzentrieren wir uns auf Europa und die europäischen Vorschriften.

Im Weiteren ist dieser Betrag wie folgt gegliedert: Zunächst geben wir einen Überblick über die Voraussetzungen (Abschnitt 2). Danach wird in Abschnitt 3 das vorgeschlagene Schema für die Ableitung einer Identität mithilfe von PSD2 vorgestellt. Abschnitt 4 bewertet unser Schema nach eIDAS LoA [Eu15] und gibt einen Überblick über das mit unserem Schema erreichbare Maß an Sicherheit. Verwandte Arbeiten werden in Abschnitt 5 besprochen. Abschließend fassen wir diese Arbeit zusammen und zeigen zukünftige Schritte auf (Abschnitt 6).

2 Voraussetzungen

IT-Systeme im öffentlichen Sektor müssen staatliche Vorgaben und Richtlinien berücksichtigen. Diese Vorgaben werden von der Europäischen Union (EU) herausgegeben und die Staaten setzen diese bedarfsgerecht um. Je nach Land muss ein IT-System unterschiedliche Anforderungen erfüllen.

Wir werden wichtige Voraussetzungen überprüfen, die unser Schema berücksichtigen muss.

eIDAS Die eIDAS-Verordnung ist ein gemeinsamer Rahmen, der Standards für elektronische Identifizierungs- und Treuhanddienste für elektronische Transaktionen auf dem europäischen Markt festlegt. Mit dieser Verordnung sollen die Voraussetzungen für eine grenzüberschreitende Online-Authentifizierung mit nationalen elektronischen Ausweisen geschaffen werden (vgl. [Eu14]).

Vertrauensniveaus Die Vertrauensniveaus nach der eIDAS-Verordnung definieren das Vertrauen, das in einen bestimmten Mechanismus gesetzt werden kann. Hierzu werden die Anforderungen beschrieben, die erfüllt sein müssen, um das angestrebte Niveau zu erreichen. Die eIDAS-Verordnung definiert drei Vertrauensniveaus: *hoch*, *substantiell* und *niedrig*. Je höher das Vertrauensniveau, desto mehr Anforderungen müssen erfüllt werden.

PSD2 Die Payment Service Directive 2 (PSD2) ist eine Richtlinie der EU, welche u. a. eine Schnittstelle bei Zahlungsdienstleistern (z. B. Banken) vorschreibt (vgl. [EB17]). Sie wurde für die Öffnung des Bankenmarktes für neue Unternehmen im Bereich der Zahlungsdienste geschaffen. Die Zahlungsdienstleister müssen Dritten Zugang zu ihren Kontodaten und -funktionen gewähren. Dies erfordert immer die Erlaubnis des Kontoinhabers. Die Informationen, die laut Richtlinie zur Verfügung gestellt werden müssen, sind:

- Kontoinformationen

- Leistungsbilanzsaldo
- Transaktionen der letzten 90 Tage

Für die Ableitung der Identität werden vom PSD2-Interface nur die Kontoinformationen benötigt (vgl. Abschnitt 3). Diese Kontoinformation stellen die Identität dar, welche abgeleitet werden soll. Zwei-Faktor-Authentifizierung ist für die PSD2-Authentifizierung obligatorisch, aber es ist auch möglich, mehr als zwei Faktoren zu verwenden. Die Authentifizierung eines Benutzers ist nur für eine Aktion gültig, z. B. den Zugriff auf die Kontoinformationen. Um dies zu erreichen, wird aus der Zwei-Faktor-Authentifizierung ein Authentifizierungscode generiert, der dann der Aktion zugeordnet wird. Wenn der Benutzer z. B. die Summe einer Transaktion ändern will, wird dies als neue Aktion betrachtet und es muss ein neuer Authentifizierungscode erzeugt und verwendet werden. Der Authentifizierungscode wird im letzten Schritt verwendet, um die Ausführung der zugehörigen Aktion zu autorisieren.

3 Identitäten ableiten mithilfe der PSD2

Unser Schema basiert auf dem OAuth 2.0 Protokoll. OAuth 2.0 gilt formal als sicher (vgl. [FKS16]) und unterstützt Zwei-Faktor-Authentifizierung (vgl. [Ha12]). Es eignet sich daher gut, um den hohen Anforderungen an das Vertrauensniveau der PSD2 gerecht zu werden. Eine gemeinsame API für Banken würde den Implementierungsaufwand erheblich vereinfachen, da nicht jede API einzeln zur Anwendung hinzugefügt werden muss. Es gibt bereits Zusammenschlüsse von Banken um in Zukunft eine gemeinsame API für PSD2 zur Verfügung stellen zu können (z. B. Preta[5] und der Berlin Group PSD2 Standard[6]). Neben diesen Zusammenschlüssen gibt es noch APIs von einzeln Banken. Die in der von uns entwickelten Android App zu Demonstrationszwecken genutzte API ist die der Deutschen Bank (dbAPI) [De17] (Abbildungen der App befinden sich im Anhang auf Seite 181). Diese wurde gewählt, da sie auf dem OAuth 2.0 Protokoll aufbaut und bereits öffentlich zur Verfügung steht.

Abbildung 2 zeigt den Nachrichtenfluss für die Ableitung einer Identität. Um persönliche Informationen eines Benutzers von einem Account Servicing Payment Service Provider (ein kontoführender Zahlungsdienstleister, siehe [EB17] Artikel 4, Nr. 17) (ASPSP) zu erhalten, benötigen wir eine Berechtigung, um auf die erforderlichen Informationen zugreifen zu können. Unser Schema erlaubt es dem Benutzer, den IdP zu autorisieren und auf die Benutzerdaten des ASPSPs zuzugreifen. Abbildung 2 zeigt drei Entitäten. *ASPSP* repräsentiert die API der Deutschen Bank. Diese könnte auch durch eine andere Serviceschnittstelle eines ASPSPs ersetzt werden. *aIdP* ist der IdP, bei dem die abgeleitete Identität nach dem Ableitungsprozess gespeichert wird. Benutzer / Kontoinhaber stellt zwei Dinge dar: Ein Benutzer, der eine Identität zum aIdP ableiten will und ein Kontoinhaber, der über die

[5] Petra `https://www.preta.eu/`

[6] Berlin Group PSD2 Standard `https://www.berlin-group.org/psd2-access-to-bank-accounts`

PSD2-Schnittstelle auf ein Bankkonto zugreifen kann. Benutzer und Kontoinhaber sind ein und dieselbe Person. Folgende Daten werden von der Deutschen Bank bereit gestellt und müssen vorliegen, bevor die Ableitung durchgeführt werden kann:

- **App Client ID**: Eine eindeutige ID, welche der ASPSP registrierten Anwendungen zur Verfügung stellt. Mit dieser soll der Server, welcher später die Anfragen an den ASPSP stellt, identifiziert werden. Diese ID wird der Software auf dem aIdP beim Start als Parameter übergeben.
- **App Client Secret Key**: Der vom ASPSP festgelegte geheime Schlüssel der Anwendung, mit dem diese ihre Authentizität gegenüber dem ASPSP nachweisen kann. Auch dieser Schlüssel wird der Software auf dem aIdP beim Start als Parameter übergeben.

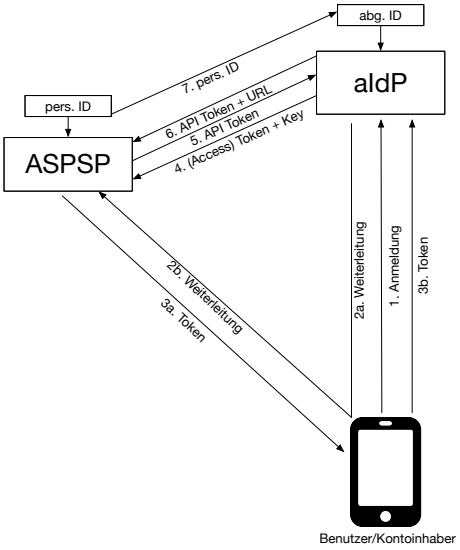


Abb. 2: Ableitung von Identitäten mithilfe der PSD2

Für die Ableitung der Identität werde folgende Schritte durchgeführt:

1 Der Benutzer meldet sich beim aIdP an. Es spiele keine Rolle, ob hierzu eine Zwei-Faktor-Authentifizierung oder der Benutzername mit einem Passwort genutzt wird, da die Ableitung selbst sicherstellt, dass die benötigten Identitätsdaten nur dann transferiert werden, wenn der Benutzer dazu berechtigt ist. Nach der Anmeldung initiiert der Benutzer die Ableitung.

2a Der aIdP leitet den Benutzer an den ASPSP weiter, bei dem der Benutzer ein Konto besitzt. Hierbei wird die App Client ID und eine Statusvariable mitgesendet. Diese

enthält, verschlüsselt mit einem Schlüssel des aIdP, das momentane Session-Cookie des Benutzers und eine nicht leicht zu erratende Zeichenkette (möglichst lang und zufällig). Dieses Vorgehen führt zu einer Bindung der Session an das momentan verbundene Smartphone des Benutzers. Daher sind eventuell abgefangene Daten für einen Identitätsdiebstahl am aIdP nicht zu gebrauchen.

2b Der Benutzer authentifiziert sich als Kontoinhaber gegenüber dem kontoführenden Zahlungsdienstleister mit seinen Zugangsdaten. Da PSD2 Zwei-Faktor-Authentifizierung erfordert, ist es zwingend erforderlich, einen zweiten Faktor für die Authentifizierung anzugeben. Bei ASPSPs handelt es sich dabei in der Regel um eine sichere TAN-Methode.

3a Nach erfolgreicher Authentifizierung sendet der ASPSP einen einmaligen Authentifizierungscode (hier Token gennannt) an den Kontoinhaber. Damit ist es möglich, gegenüber dem ASPSP zu beweisen, dass der Kunde den Zugriff auf die Kontodaten autorisiert hat. Der Token wird mit dem Kunden verknüpft, so dass mit diesem Token nur die Kontodaten dieses einen Kunden abgerufen werden können. Zusätzlich wird die App Client ID an den Token gebunden, so dass dieser Token nur von dem autorisierten Service (dem aIdP) genutzt werden kann.

3b Anschließend wird der Token automatisch vom Smartphone des Benutzers an den aIdP übertragen. Zusätzlich wird die Statusvariable mitgesendet, um die Bindung des Tokens mit der Session des Benutzers sicherzustellen. Wir gehen davon aus, dass, wenn der Benutzer in der Lage ist, einen Token für den Zugriff bereitzustellen, er auch der Kontoinhaber ist. Unter dieser Annahme sind die Daten des Kontos des ASPSPs die Identität des Benutzers.

4 Der aIdP authentisiert sich beim ASPSP mit dem *App Client Secret Key* und dem vom Benutzer übermittelten Token, um den Zugriff auf die Kontodaten des Kunden zu legitimieren.

5 Nach der Authentifizierung wird ein API Token an den aIdP gesendet. Dieser ermöglicht es, dem aIdP für einen eingeschränkten Zeitraum auf die vom Benutzer / Kontoinhaber autorisierten Kontodaten zuzugreifen.

6 Der aIdP fordert nun durch einen Zugriff auf die API die Kontodaten des Benutzers an. Hierzu überträgt er seinen persönlichen API Token und die URL zu der gewünschten Ressource.

7 Der ASPSP Server überträgt die angeforderten persönlichen Identitätsdaten (pers. ID). In unserem Fall sind dies folgende Daten: Anschrift (Straße, Hausnummer, Postleitzahl, Ort, Land, Eingetragener Wohnsitz), Persönliche Daten (Vorname, Nachname, Geburtsdatum, Geschlecht, Akademischer Titel, Adelstitel, Nationalität, Geburtsname, Geburtsort, Geburtsland, E-Mail-Adressen, Telefonnummern, Internationale Vorwahl, Ortsnetzkennzahl) und Legitimation des Kunden gegenüber des ASPSP (Dokumenttyp (Reisepass, ID,...), Dokumentnummer, Ausgabedatum des Dokuments,

Ausstellende Behörde, Ablaufdatum). Es werden nicht alle Daten benötigt. Daher sollte aus Gründen der Datensparsamkeit nur die benötigten Daten übernommen werden. Diese können je nach Einsatzzweck des aIdP unterschiedlich ausfallen.

Dieses Schema hat die folgenden Eigenschaften: Der Anwender benötigt keinen Kartenleser oder ähnliche Hardware. Dies schränkt den Benutzerkreis wegen fehlender Hardware nicht ein. Außerdem ist es nicht erforderlich, dass der Benutzer eine elektronischen Identitätsausweis (in Deutschland einen nPA) besitzt. Es genügt, dass sich der Benutzer in der Vergangenheit erfolgreich bei einem ASPSP registriert hat (was nach nationalem Recht ein sichererer Identitätsnachweis ist). Die potentielle Benutzergruppe des Systems ist somit jede Person mit einem amtlichen Ausweis, wie z. B. Reisepass oder dem alten, nicht-digitalen Personalausweis und ein Bankkonto bei einer europäischen bzw. teilnehmenden Bank. Darüber hinaus kann die Schnittstelle des kontoführenden Zahlungsdienstleisters für den Kontoinhaber (Schritt 2b. in Abbildung 2) so gestaltet werden, wie es der Kontoinhaber von seinem kontoführenden Zahlungsdienstleister gewohnt ist (z. B. Online-Banking). Dies erleichtert dem Benutzer den Zugriff auf das System, da er in der Regel die notwendigen Schritte zur Authentifizierung bereits vom Online-Banking kennt.

Bevor der Benutzer auf die abgeleitete Identität zugreifen kann, wird das Smartphone des Benutzers mit der abgeleiteten Identität verknüpft. Diese muss mit einer Zwei-Faktor-Authentifizierung abgesichert werden. Einer der Faktoren im System ist das Smartphone, das einen privaten Schlüssel speichert, der auf dem Smartphone generiert wurde. Dieser private Schlüssel wird für die Authentifizierung beim aIdP verwendet. Der zweite Faktor kann vom Benutzer, je nach Möglichkeiten des Smartphones, ausgewählt werden. Der Faktor kann z. B. eine PIN oder ein Fingerabdruck sein.

Der aIdP kann, nach der Ableitung, in einem System als Single Sign-on (SSO) IdP genutzt werden, da die Identitäten der Benutzer bereits vorhanden und der Zugriff auf diese durch Zwei-Faktor-Authentifizierung geschützt sind. Hierzu müssten Technologien wie OAuth 2.0 oder SAML genutzt werden, um die entsprechende Funktionalität zu implementieren. Aus Datenschutzgründen und der Menge der möglicherweise vorhanden Benutzerdaten, ist es empfehlenswert hierbei den Zugriff für andere Dienste zu beschränken und auch die Unterstützung von „Assertions“ (z. B. „ist der Benutzer schon volljährig?“) zu bedenken. Damit der aIdP später beweisen kann, dass er die Daten tatsächlich von der Bank erhalten hat und diese unverändert sind, kann eine Signatur des ASPSP erforderlich sein. Dies ist jedoch von PSD2 nicht vorgesehen, könnte jedoch für unser Schema eine hilfreiche Erweiterung sein.

4 Evaluation des Vertrauensniveaus

Für die Bewertung des Vertrauensniveaus wird eIDAS LoA [Eu15] verwendet. Diese Richtlinie definiert die Anforderungen an die Einhaltung der in Europa geltenden Vertrauensniveaus. Im Folgenden werden die Ergebnisse der Evaluierung für die einzelnen

Bereiche der Richtlinie präsentiert. Des weiteren wird gezeigt, was getan werden muss, um das Vertrauensniveau *substantiell* zu erreichen. Für die Zwei-Faktor-Authentifizierung beim aIdP schlagen wir FIDO UAF vor, um den Zugriff auf die abgeleitete Identität zu sichern (vgl. [LBH15]).

Beantragung und Eintragung Die Sicherheit der Anmeldung hängt von der Authentifizierung des kontoführenden Zahlungsdienstleisters ab. Die Richtlinien des NIST [Gr17] enthalten sichere TAN Verfahren, welche bei einer Zwei-Faktor-Authentifizierung genutzt werden können. Die Verfahren photoTAN und chipTAN erreichen *substantiell*. SMS/Mobile-TAN werden noch von einigen nationalen Richtlinien bei bereits bestehender Software akzeptiert (z. B. BSI TR-03107-1 [BS16]), sollen aber nicht mehr bei neuen Anwendungen zum Einsatz kommen. TAN und iTAN sind nicht sicher. Die Architektur erreicht in diesem Punkt das Vertrauensniveau *substantiell*.

Identitätsnachweis und -überprüfung Der Identitätsnachweis und die Identitätsüberprüfung erfolgt beim ASPSP bereits vor der Ableitung. ASPSPs sind verpflichtet eine Identitätsfestellung auf Vertrauensniveau *hoch* durchzuführen [EB17]. Daher kann davon ausgegangen werden, dass dies durchgeführt wird, bevor ein Kunde ein Konto eröffnen kann.

Merkmale und Gestaltung elektronischer Identifizierungsmittel Hier kommt es darauf an, wie gut der private Schlüssel durch das Smartphone geschützt ist. Vertrauensniveau *hoch* kann in diesem Bereich nicht erreicht werden, da nicht jedes Smartphone einen „Schutz vor Duplizierung und Fälschung vor Angreifern mit hohem Angriffspotential“ bietet (vgl. [Gr17]). Das Vertrauensniveau *substantiell* wird hingegen erreicht, da jedes iOS oder Android Smartphone mindestens auf Softwareebene den Zugriff auf private Schlüssel schützt (vgl. [Go17], [Ap17]).

Ausstellung, Auslieferung und Aktivierung Die abgeleitete ID ist durch die zwei Faktoren von FIDO gesichert. Diese zwei Faktoren werden auf dem Smartphone des Benutzers erzeugt / festgelegt, daher besteht keine Gefahr, dass diese bei der Erstellung in den Besitz einer unberechtigte Person kommen. Dies erfüllt die Anforderungen für das Vertrauensniveau *substantiell*.

Aussetzung, Widerruf und Reaktivierung Architektur erlaubt einen Widerruf, wenn der Benutzer im Besitz seiner Authentifizierungsmittel ist. Ist dies nicht der Fall, besteht keine Möglichkeit, die abgeleitete Identität zu widerrufen. Um das Vertrauensniveau *substantiell* zu erreichen, wird noch ein anderer Weg benötigt, die abgeleitete Identität zu widerrufen. Dies kann bspw. erreicht werden, wenn der Benutzer unter Anwendung von Benutzername

und Passwort seine FIDO Verknüpfung löscht. Ebenso erfüllt eine 24h Telefon-Support-Hotline die Anforderungen. Anschließend müssen bei beiden Methoden auch alle anderen Benutzerdaten auf dem gleichen Vertrauensniveau gelöscht werden. Aufgrund des Fehlens eines zweiten Faktors erreichen beide Transaktionen kein *substanzielles* Vertrauensniveau. Dieses Verfahren stellt jedoch sicher, dass die personenbezogenen Daten gelöscht und somit nicht missbräuchlich verwendet werden können.

Verlängerung und Ersetzung Das Ablaufdatum, welches bei der Ableitung vom ASPSP an den aIdP übertragen wurde, ermöglicht es, die abgeleitete ID des Benutzers zu sperren. Danach muss der Kunde eine erneute Ableitung anstoßen, um so eine neue abgeleitete ID zu erhalten. Die Verlängerung erfolgt somit auf dem gleichen Vertrauensniveau, wie bei der erstmaligen Ableitung und erfüllt die Anforderungen an das Vertrauensniveau *substantiell*.

Authentifizierung bei dem aIdP Die beiden Faktoren und das Authentisierungsprotokoll erfüllen die Anforderungen für das Vertrauensniveau *substantiell*. FIDO erfüllt die Anforderungen an die geforderte dynamische Authentifizierung (vgl. [LBH15]). Die Faktoren werden über keinen Kanal gesendet, sondern auf dem Smartphone erzeugt / festgelegt. Die Übertragung der Daten erfolgt, wie im gesamten Schema, über TLS. Dabei sind die entsprechenden Empfehlungen von OWSAP [7] und für Deutschland die BSI-Richtlinien (vgl. [BS18]) für den Einsatz von TLS zu berücksichtigen.

Management und Organisation Das Erreichen des Vertrauensniveau *substanziell* hängt von der Organisation und dem Management des aIdP ab. Dies kann nicht im Rahmen dieser Arbeit evaluiert werden, da die Bewertung von der jeweiligen Institution abhängt, welche das Schema implementieren möchte. Dennoch werden folgend ein paar wichtige Punkte hervorgehoben. Es ist eine Überprüfung nach ISO27001 [IS13] (oder ähnlich) erforderlich. Jeder kontoführende Zahlungsdienstleister erfüllt die Anforderungen für das Vertrauensniveau *substantiell*, da bei diesen eine Überprüfung nach ISO27001 (oder ähnlich) vorgeschrieben ist (vgl. [Eu13]).

Zusammengefasst erfüllt unser Schema die Voraussetzungen für das Vertrauensniveau *substantiell*, wenn bestimmte Anforderungen in dem System erfüllt sind, in das unser Schema implementiert werden soll. Insbesondere muss darauf geachtet werden, dass das System es erlaubt, die abgeleitete Identität zu widerrufen, auch wenn der Benutzer nicht mehr auf sein Smartphone zugreifen kann. Außerdem müssen alle teilnehmenden Institute nach ISO27001 überprüft worden sein.

[7] OWSAP - Transport Layer Protection Cheat Sheet `https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet#Server_Protocol_and_Cipher_Configuration`

5 Verwandte Arbeiten

Es gibt, neben Lösungen einzelner Banken, auch APIs, welche von mehreren Bank eingesetzt werden. Unter anderem ist dies *BankId* (vgl. [Ba17]), welche den Zugriff auf alle Bankkonten norwegischer Banken ermöglicht und sich bereits im Einsatz befindet. Ebenso ist *Open Banking Ltd.* (vgl. [SNS17]) bereits in Großbritannien in Verwendung. Eine API, welche sich nicht auf ein Land konzentriert, ist die *OpenID Financial API (FAPI) WG* (vgl. [Op17]). Deren Spezifikationen sind noch nicht final, befindet sich aber bereits im Draft Status. Alle drei Lösungen können als Vorentwicklung zu einer PSD2 API Schnittstelle gesehen werden. Ebenso können sie, bei einem ASPSP implementiert, in unserem Schema verwendet werden.

Die Arbeit „Auf dem Weg zur Umsetzung der PSD2-Richtlinie“ von Hühnlein et al. gibt einen Überblick über die Anforderungen an eine PSD2 API. Dabei werden insbesondere die Anforderungen, welche durch die verschiedenen EU Verordnungen entstehen, analysiert. Ebenso wird die Sicherheit der möglicherweise beim ASPSP eingesetzten Technologien (OAuth, SAML) untersucht (vgl. [Hu17]). Im Gegensatz hierzu untersucht die vorliegende Arbeit das Vertrauensniveau einer Ableitung von Identitätsdaten in einen aIdP.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Schema zur Ableitung einer Identität mittels PSD2 vorgeschlagen, welches das Vertrauensniveau *substantiell* erreicht. Hierzu müssen jedoch weitere Anforderungen vom System erfüllt werden, in das dieses Schema implementiert werden soll. Insbesondere muss darauf geachtet werden, dass ein System es erlaubt, die abgeleitete Identität zu widerrufen, auch wenn der Benutzer nicht mehr auf sein Smartphone zugreifen kann. Zusätzlich erwarten wir eine hohe Benutzerakzeptanz, da der Anwender mit der Oberfläche und der Authentifizierungsmethode vertraut ist, die von seinem kontoführenden Zahlungsdienstleister verwendet wird. Ohne den Einsatz eines digitalen Personalausweises wird eine größere Anzahl potenzieller Benutzer angesprochen. Um die volle Leistungsfähigkeit zu erreichen, müssen bei der Implementierung unseres Schemas die folgenden Anforderungen an ein Systemdesign berücksichtigt werden: die kontoführende Zahlungsdienstleister müssen Authentisierung auf Sicherheitsniveau substantiell unterstützen. Jede beteiligte Institution des Systems muss ein Überprüfung nach ISO27001 (oder ähnlich) vorweisen. Zusätzlich ist eine Widerrufsmöglichkeit erforderlich.

In einem nächsten Schritt wollen wir unseren Prototypen verbessern und erste Usability-Tests an diesem durchführen. In diesem Zusammenhang werden wir eine PSD2-Spezifikationserweiterung für die Ableitung von Identitäten vorschlagen.

7 Danksagung

Diese Arbeit wurde vom Hessischen Ministerium für Inneres und Sport (HMdIS) im Rahmen der Förderung “Runder Tisch Cybersecurity@Hessen"gefördert.

Literaturverzeichnis

[Ap17] Apple Inc.: iOS Security. 2017. https://www.apple.com/business/docs/iOS_Security_Guide.pdf, Zugriff am: 13.12.2017.

[Ba17] BankID Norge AS: BankID. 2017. https://www.bankid.no/en/, Zugriff am: 13.12.2017.

[BS16] BSI: Elektronische Identitäten und Vertrauensdienste im E-Government – Teil 1: Vertrauensniveaus und Mechanismen. Bericht 1.1, Bundesamt für Sicherheit in der Informationstechnik, Bonn, DE, Oktober 2016. The English translation *Technical Guideline TR-03107-1 V1.0 Electronic Identities and Trust Services in E-Government Part 1* is outdated.

[BS18] BSI: Kryptographische Verfahren: Empfehlungen und Schlüssellängen – Teil 2 – Verwendung von Transport Layer Security (TLS). Bericht, Bundesamt für Sicherheit in der Informationstechnik, Bonn, DE, Januar 2018.

[Bu11] Burr, William E.; Dodson, Donna F.; Newton, Elaine M.; Perlner, Ray A.; Polk, W. Timothy; Gupta, Sarbari; Nabbus, Emad A.: SP 800-63-1. Electronic Authentication Guideline. Bericht, Gaithersburg, MD, United States, 2011.

[De17] Deutsche Bank: Deutsche Bank API Program. 2017. https://developer.db.com, Zugriff am: 13.12.2017.

[EB17] EBA: Draft Regulatory Technical Standards – on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2). Standard, European Banking Authority, Februar 2017.

[Eu13] European Union: DIRECTIVE 2013/36/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL 910/2014 on access to the activity of credit institutions and the prudential supervision of credit institutions and investment firms, amending Directive 2002/87/EC and repealing Directives 2006/48/EC and 2006/49/EC. Juni 2013.

[Eu14] European Union: COMMISSION IMPLEMENTING REGULATION (EU) 910/2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Juli 2014.

[Eu15] European Union: COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market. September 2015.

[FKS16] Fett, Daniel; Küsters, Ralf; Schmitz, Guido: A Comprehensive Formal Security Analysis of OAuth 2.0. ACM Press, S. 1204–1215, 2016.

[Go17] Google: Android Keystore System. 2017. https://developer.android.com/training/articles/keystore.html, Zugriff am: 13.12.2017.

[Gr17] Grassi, Paul A; Newton, Elaine M; Perlner, Ray A; Regenscheid, Andrew R; Burr, William E; Richer, Justin P; Lefkovitz, Naomi B; Danker, Jamie M; Choong, Yee-Yin; Greene, Kristen et al.: NIST Special Publication 800-63B: Digital identity guidelines: authentication and lifecycle management. Bericht, NIST, 2017.

[Ha12] Hardt, Dick: RFC 6749: The OAuth 2.0 Authorization Framework. 2012. https://tools.ietf.org/html/rfc6749, Zugriff am: 13.12.2017.

[Hu17] Huehnlein, Detlef; Hühnlein, Tina; Wich, Tobias; Nemmert, Daniel; Rauh, Michael; Baszanowski, Stefan; Prechtl, Mike; Lottes, René: Auf dem Weg zur Umsetzung der PSD2-Richtlinie. In: D-A-CH Security 2017 Tagungsband. syssec, 2017.

[IS13] ISO: Information technology – Security techniques – Information security management systems – Requirements. Standard, International Organization for Standardization, Geneva, CH, 2013.

[LBH15] Lindemann, Rolf; Baghdasaryan, Davit; Hill, Brad: FIDO Security Reference. FIDO Alliance Proposed Standard, 2015.

[Op17] Open Banking Ltd: Read/Write APIs, Version 1.0.0. 2017. https://www.openbanking.org.uk/read-write-apis/, Zugriff am: 13.12.2017.

[Po16] Poushter, Jacob: Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies. Pew Research Center's Global Attitudes Project, Februar 2016. http://www.pewglobal.org/2016/02/22/smartphone-ownership-and-internet-usage-continues-to-climb-in-emerging-economies/, Zugriff am: 12.12.2017.

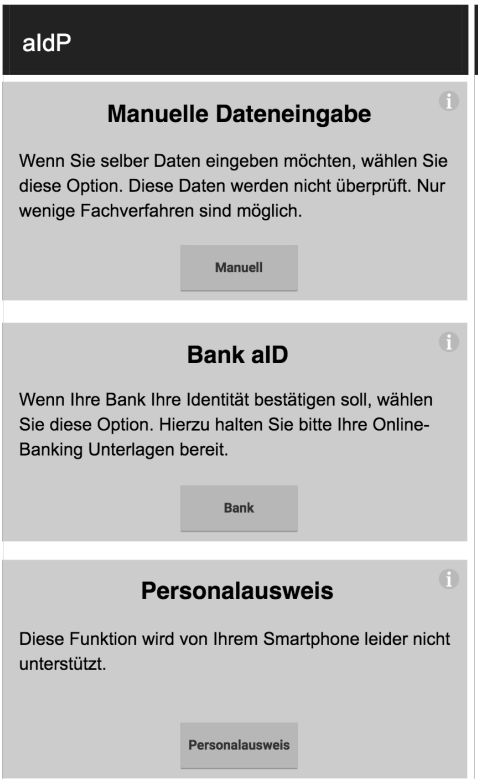
[rB16] 81 % der Internetnutzer gehen per Handy oder Smartphone ins Internet, https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2016/12/PD16_430_63931.html, Zugriff am: 13.12.2017.

[SNS17] Sakimura, Nat; Nadalin, Tony; Saxena, Anoop: OpenID Financial API (FAPI) WG. 2017. https://openid.net/wg/fapi/, Zugriff am: 13.12.2017.

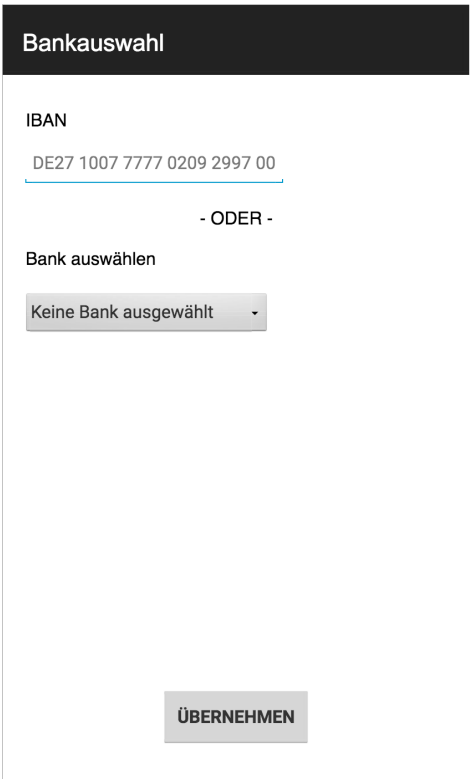
[Va15] Vaida, Laura: Akzeptanz von E-Government-Anwendungen in Deutschland. 2015.

[WHM16] Willomitzer, Jörg; Heinemann, Andreas; Margraf, Marian: Zur Benutzbarkeit der AusweisApp2. Mensch und Computer 2016–Workshopband, 2016.

Anhang

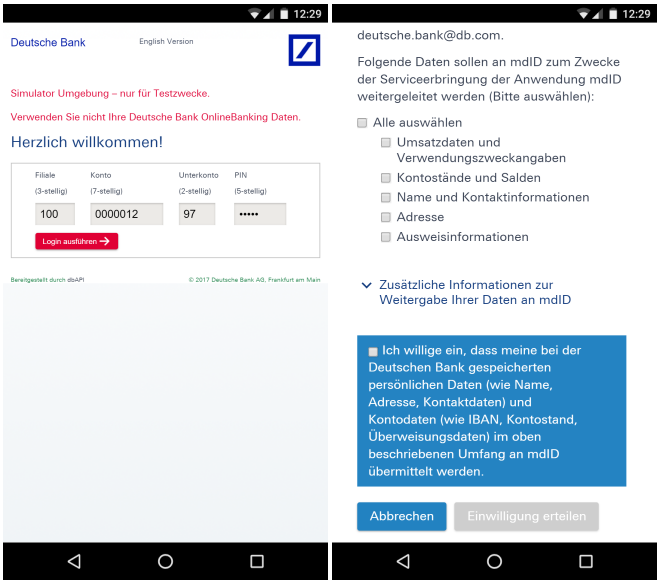


(a) Mit einem Klick auf Bank wird der Ableitungsprozess gestartet



(b) Die Bank kann über die IBAN oder ein Auswahlfeld gewählt werden. Im Demonstrator ist zur Zeit nur die Deutsche Bank verfügbar.

Abb. 3: Android App: Ableitung der Identität, Teil 1



(a) Der Benutzer authentifiziert sich. (b) Auswahl, welche Daten an den aIdP übertragen werden dürfen.

Abb. 4: Android App: Ableitung der Identität, Teil 2

(a) Übersicht über die abgeleiteten Daten.

Abb. 5: Android App: Ableitung der Identität, Teil 3

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 197

Comparative Usability Evaluation of Cast-as-Intended Verification Approaches in Internet Voting

Karola Marky[1], Oksana Kulyk[2], Melanie Volkamer[3 4]

Abstract:

Internet Voting promises benefits like the support for voters from abroad and an overall improved accessibility. But it is accompanied by security risks like the manipulation of votes by malware. Enabling the voters to verify that their voting device casts their intended votes is a possible solution to address such a manipulation - the so-called *cast-as-intended verifiability*. Several different approaches for providing cast-as-intended verifiability have been proposed or put into practice. Each approach makes various assumptions about the voters' capabilities that are required in order to provide cast-as-intended verifiability. In this paper we investigate these assumptions of four chosen cast-as-intended approaches and report the impact if those are violated. Our findings indicate that the assumptions of cast-as-intended approaches (e.g. voters being capable of comparing long strings) have an impact on the security of Internet Voting systems. We discuss this impact and provide recommendations how to address the identified assumptions and give important directions in future research on usable and verifiable Internet Voting systems.

Keywords: E-Voting; Cast-as-Intended Verifiability; Usability Evaluation

1 Introduction

Internet Voting can offer many benefits. Voters from abroad are easily integrated into an election and the election overall has a better accessibility. Although these benefits can be appealing for governments and private organizations planning to introduce Internet Voting, it also introduces new risks that are not present in traditional paper-based voting. One of those risks lies in the manipulation of votes during vote casting. The votes are cast in an unsupervised environment, in which the voters use their own personal devices. Malware on the voters' personal devices cannot be ruled out.

Hence, if an adversary wants to maliciously influence the election outcome, he or she could take control over the voting devices via malware. This malware could replace votes cast by the voters with votes for the adversary's preferred candidate. Enabling the voters to

[1] Technische Universität Darmstadt, Deutschland, karola.marky@secuso.org
[2] Technische Universität Darmstadt, Deutschland, oksana.kulyk@secuso.org
[3] Karlsruhe Institute of Technology, Deutschland, melanie.volkamer@kit.edu
[4] Technische Universität Darmstadt, Deutschland

doi:10.18420/sicherheit2018_15

verify that their voting device casts the intended vote is a possible solution to address such a manipulation - the so-called *cast-as-intended verifiability*. The concept of cast-as-intended verifiability is an aspect of *end-to-end verifiability*[5] which is the possibility to verify the integrity of the election result through all stages of an election. However, as opposed to other kinds of verification encompassed by end-to-end verifiability, existing cast-as-intended approaches require active voter involvement. Therefore, usability plays an important role in ensuring that the verification can be carried out successfully. An important aspect of usability is ensuring that the voters possess the necessary capabilities required to carry out the verification process. Assumptions on such voter capabilities are often implicitly made by the designers of cast-as-intended approaches. These assumptions and their impact on the cast-as-intended verification have, to our knowledge, not been systematically investigated yet.

In this paper we investigate the assumptions on voter capabilities of four cast-as-intended approaches in an expert evaluation. We identify steps that require voter interaction and give a detailed explanation of each investigated approach from the voter's perspective. For each step we report the assumptions about voter capabilities as well as the impact if the assumptions are violated. Our investigation indicates that the violation of assumptions can be exploited by adversaries to manipulate the election and therefore compromise the Internet Voting system's security. Furthermore, our investigation identifies security-critical assumptions on voter capabilities that cannot be expected to hold among the general voters population. Such capabilities include voters being able to compare long random sequences of characters or having access to multiple computational devices. It follows that the capabilities of voters should not be underestimated and should be taken into account when choosing and/or developing an Internet Voting system. Therefore, our work is a stepping stone into a holistic evaluation and comparison of different cast-as-intended approaches.

2 Evaluation Methodology

Several attempts to classify available cast-as-intended verification approaches have been made in the literature [GC16, Pu17, KW17]. The classification by Guasch [GC16] is the most complete available in the literature and considers the following five categories: (1) *challenge or cast*, (2) *decryption-based*, (3) *verifiable optical scanning*, (4) *verification with codes* and (5) *hardware-based verification*. For our evaluation we decided to investigate one approach per category. The category *verifiable optical scanning* does not contain approaches for Internet Voting and was therefore dropped.

The goal of our work was to evaluate the usability of the approaches independent of the user-interface. While implementing the proposed approaches and conducting usability studies

[5] The other parts are recorded-as-cast (the recorded vote does not differ from the cast vote, tallied-as-recorded (the recorded vote is correctly included in the tally) and eligibility verifiability (only votes cast by eligible voters are included in the tally) [Ad06, GV10].

is a more traditional approach, it also has drawbacks. The choice of the user-interface has an impact on the usability of the solution. In an election the user-interface is often adjusted to the specific electorate and election setting. While the question of developing the most usable user-interface for a given verifiability method is important, it hampers evaluating and comparing the usability of the approaches themselves. The approaches play an important role in usability, in case they make assumptions on voter capabilities (e.g. by requiring the voter to compare character sequences of a given length). Hence, a user-interface-independent evaluation is a vital step in deciding whether a verifiability approach is usable enough to be used in an election. Therefore, we also do not cover user-interface-based assumptions (e.g. the visibility and position of buttons or the understandability of instructions), because of its dependence from the specific user-interface.

Our method is based on the so-called *cognitive walkthrough* [Po92]. The original method targets the evaluation of user-interfaces. We adjusted the method to carry out the evaluation independent from the user-interface. We considered the steps that require interaction from the voters attempting to verify their votes (i.e. that cannot be automated by delegating them to a voting client or supporting software). Based on these steps we performed a modified cognitive walkthrough with the following questions: (a) Which assumption about voter capabilities is made? and (b) What is the impact if the assumption does not hold? If available, we relied on the literature in the field human-computer interaction that provides insight of user capabilities to perform a given task. The process was conducted by two authors of the paper independently from each other. The findings were discussed and a final assumption and impact for each step of the approach was agreed upon.

3 Results

During our evaluation we collected the following assumptions on voter capabilities in the investigated approaches. We give an overview on the assumptions first and subsequently describe the approaches. The assumptions made by the investigated approaches are:

A1: Entering. The voter enters a value without errors.
A2: Storing. The voter stores a value without errors.
A3: QR Scan. The voter is capable of scanning a QR code.
A4: Device. The voter has access to all devices required to carry out verification.
A5: Comparison. The voter is able to perform comparisons without errors.
A6: Access. The voter can access the component that publishes data required for verification.
A7: Search. The voter is able to search for a value (e.g. on a bulletin board).

In the evaluation we use two terms for describing the impact of an assumption. A *false positive* denotes that the voter uncovers an error or manipulation that in reality is not present. This might result in a distrust in the Internet Voting system. Although everything functioned correctly, the voters might think they have uncovered errors and might therefore loose confidence in the Internet Voting system. A possible consequence is the non-usage of

Assumption	Helios	Estonian System	Neuchâtel (2015)	Du-Vote
A1 (Entering)	-	-	✓ (Code)	✓(Code)
A2 (Storing)	-	-	-	✓ (Code)
A3 (QR Scan)	✓	✓	-	-
A4 (Device)	✓	✓	-	✓
A5 (Comparison)	✓ (Selection)	✓ (Selection)	✓ (Code)	✓ (Code)
A6 (Access)	-	-	-	✓
A7 (Search)	-	-	-	✓ (Code)

Tab. 1: Overview of assumptions by approach. ✓ denotes required, - denotes not required.

the Internet Voting system in the future. A *false negative* denotes that the voter does not uncover an error or manipulation that is present. In this case manipulation that took place is not uncovered. Therefore, the integrity of the tally result cannot be assured. Adversaries willing to maliciously influence the election result might exploit usability problems of the verification approaches and attack voters deliberately. Both cases constitute severe problems in an Internet Voting system. The second case influences the security showing that usability has an impact on it. In the first case, although security is given, it is ineffective in convincing the voters. An overview of the assumptions taken by the investigated approaches is given by Tab. 1.

3.1 Helios (Challenge or Cast)

Helios [Ad08] utilizes the so-called Benaloh Challenge [Be06, Be07] for cast-as-intended verification. Several proposals for the Benaloh Challenge that differ regarding voter interaction and security are available in the literature [Ad08, Ka11b, Ka11a, Ne14]. For our analysis we choose the approach by Neumann *et al.* [Ne14]. The approach is specifically designed to support the usage of different devices for voting and verifying. The voters are required to use a second device (e.g. a smartphone) for verifying. Therefore, it mitigates the assumption of a trusted voting device[6].

The Benaloh challenge enables verification that the voting client acts correctly via a so-called challenge or cast approach. After encrypting their selections, the voters have two options: (1) they can cast their encrypted vote or (2) they challenge the voting client by verifying whether the encrypted vote contains the intended selection. At the time of preparing an encrypted vote, a potential adversary controlling the voting client does not obtain knowledge, whether the encrypted vote will be cast or challenged. Therefore an adversary cannot be certain whether a manipulation will be undetected. We further describe the Benaloh Challenge in more details.

[6] The original Helios approach [Ad08] does neither exclude nor enforce the usage of different devices, but it is not specifically designed to support the transfer of verification data to a different device.

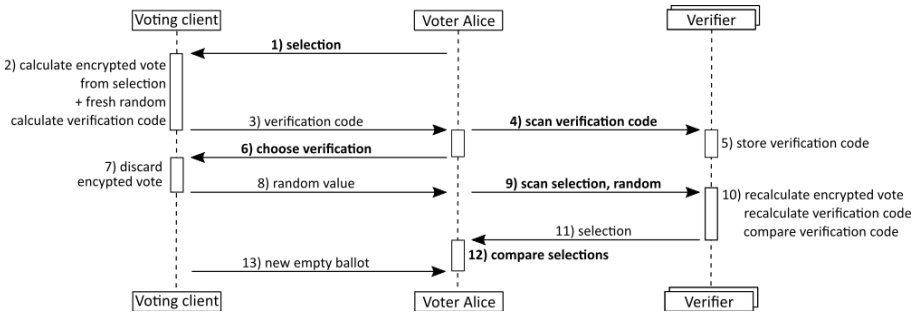


Fig. 1: Sequence diagram of one Benaloh Challenge. Steps with voter interaction are bold.

Fig. 1 provides a sequence diagram of the Benaloh Challenge with a verification device. The voter Alice commences by selecting a voting option in the voting client. The voting client encrypts Alice's selection probabilistically and derives a verification code, which is a hash from the resulting encrypted vote. Then the voting client displays this verification code to Alice. The verification code serves as a commitment of the voting client to the encrypted vote. The verification code is also displayed as QR code which Alice scans with a verification device (e.g. smartphone). Therefore, Alice needs access to such a device (A4). Otherwise, she would not able to verify at all. An adversary might be able to obtain knowledge which voters do not have access to a verification device and manipulate their votes without detection.

Alice decides whether to cast or to verify the encrypted vote. In case of verification, the voting client displays the verification data to her as a QR code. This data consists of the selection Alice previously made and the random value used during encryption. Alice scans the QR code with the verification device. Therefore, she has to be able to perform the scan (A3), otherwise she cannot verify which results in the same consequences that Alice would face by not having access to a verification device. The verification device runs an alternative software - the so-called *verifier* - to recalculate the encrypted vote. Then it derives a verification code from the encrypted vote and automatically compares it to the previously scanned one. If the verification codes match, the selection used for computation is displayed to Alice. She compares this to her previous selection. If both match, the encrypted vote contained Alice's intent, otherwise Alice can trigger an alarm. Alice is assumed to perform the comparison without errors (A5). Errors would result in a false positive (uncovering a manipulation that is not present). After verification Alice has to start from the beginning with an empty ballot[7]. If Alice chooses to cast her vote she still has to take note of the verification code by scanning the corresponding QR code. After casting, Alice's verification code is published on the election's bulletin board. In order to verify that the correct vote was cast, the verification device looks up Alice's verification code on the bulletin board. Alice is informed whether her verification code is on the bulletin board.

[7] The random value could be used for breaking Alice's vote secrecy in case she would cast this vote.

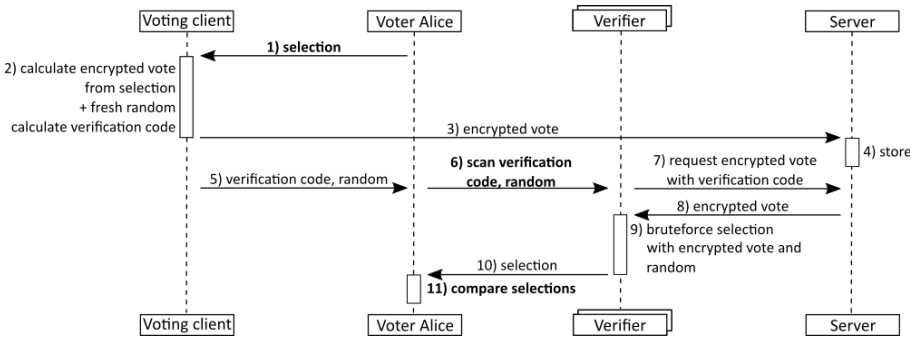


Fig. 2: Sequence diagram of verification in the Estonian system. Steps with voter interaction are bold.

3.2 Estonian System (Decryption-based)

The Estonian voting system [HW14] is a decryption-based approach. This means that the ciphertext of the encrypted vote is investigated in order to perform cast-as-intended verification. Because the ciphertext of the cast encrypted vote is inspected, vote secrecy is broken. Fig. 2 provides a sequence diagram of cast-as-intended verification in the Estonian system.

Alice opens the voting client and authenticates via an ID card. She makes her selection which is encrypted probabilistically by the voting client and subsequently send to the vote storage server. For cast-as-intended verification, the voting client generates a QR-code which contains the random value used during encryption and a hash of the encrypted vote that we denote as verification code. To verify Alice needs to install a verification application on her mobile device and scan the QR code to transfer the random value and the verification code. Therefore, Alice requires access to a mobile device (A4), otherwise she cannot verify. Alice has to be able to scan the QR code (A3), otherwise Alice cannot execute verification at all. The same consequences mentioned above in the Helios description apply here. An adversary might be able to obtain knowledge whether the voter has access to a verification device and can successfully manipulate a vote in case the voter does not have access.

The verification app uses the verification code to download the encrypted vote from the vote storage server. Using the random value, the verification app bruteforces the content of the encrypted vote by encrypting all possible voting options and comparing the resulting encrypted votes with the one downloaded from the server. Then it displays the voting option that matches the encrypted vote of Alice. She has to compare this to her previous selection without errors (A5).

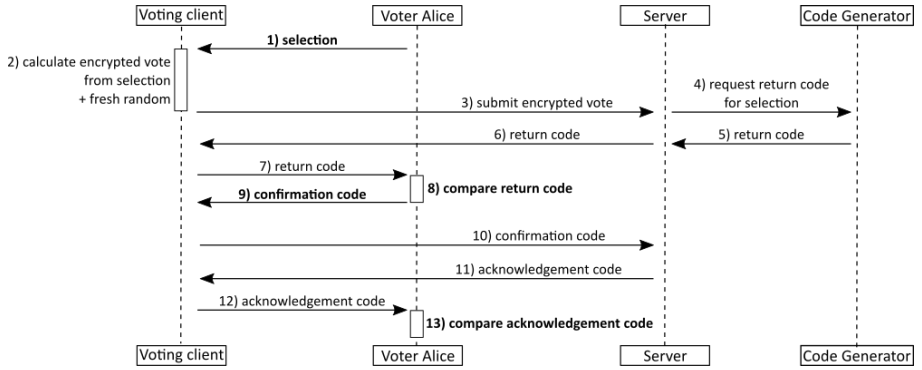


Fig. 3: Sequence diagram of verification in Neuchâtel (2015). Steps with voter interaction are bold.

3.3 Neuchâtel (2015) (Verification with Return Codes)

The cast-as-intended approach of Neuchâtel (2015) [GGP15] is based on return codes (see Fig. 3). Alice opens the voting client and indicates her selection. The selection is encrypted and sent to the voting server. The voting server contacts the code generator which answers with the return code (here called verification code) corresponding to Alice's selection. The verification code is forwarded to Alice. Prior to the election Alice has received a code sheet (e.g. via postal mail) listing her combinations of voting options and verification codes. Alice checks, whether the displayed verification code belongs to her selection. To successfully verify, she is assumed to compare the received verification code to the one on her code sheet without errors (A5). Comparing the received verification code to the wrong code on the code sheet might result in a false positive. Making an error during the comparison, even if Alice compares to the correct verification code on the code sheet, also leads to a false positive.

If the verification code is correct, Alice sends a confirmation code, that she also finds on her code sheet, to the voting server. She assumed to enter it without errors (A1), if she makes an error she cannot confirm that she compared the codes and her vote will not be included in the tally. The voting server finally responds with an acknowledgement code to confirm that it received the confirmation code. Alice is assumed to compare the received acknowledgement code without errors (A5). If she makes an error during comparison, this results in a false positive.

3.4 Du-Vote (Hardware Token)

In the Du-Vote approach [Gr15] cast-as-intended verification is based on the access to a trusted device for performing specific computations. In the case of Du-Vote the trusted device is a hardware token that Alice needs access to (A4). Otherwise she can neither vote

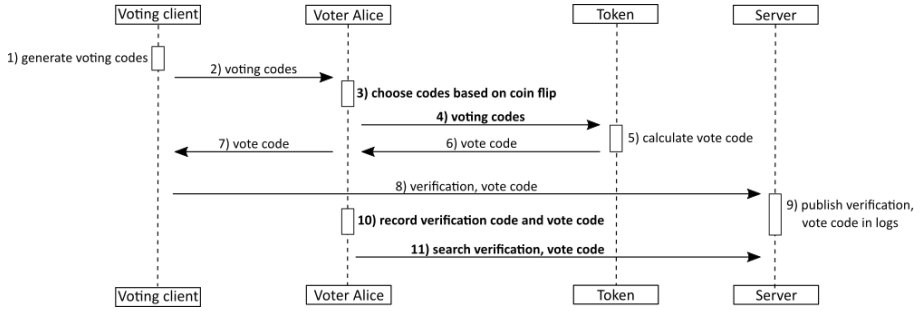


Fig. 4: Sequence diagram of verification in Du-Vote. Steps with voter interaction are bold.

nor verify. Fig. 4 provides a sequence diagram of Du-Vote. Alice gets a hardware token with unique embedded keys. Those keys are registered on the voting server. To vote Alice has to authenticate herself by previously received credentials. The voting client calculates two random values per voting option. The random values are grouped in two sets. Alice is instructed to flip a coin to randomly select one set. Now she enters all random values of the selected set into the hardware token. Then she enters the random value from the non-selected set that belongs to her selected voting option. The hardware token generates a vote code, that Alice enters in the voting client. The correctness of the verification is based on the assumption that Alice generates a correct vote code by using the hardware token. This is only the case if Alice does not make errors while entering the random values (A1). If only one wrong digit is entered, this fault propagates and leads Alice to a false negative.

Alice records her vote code as well as the vote identifier, which was generated by the voting client. To provide verification, the voting server logs all votes and publishes this data. Alice searches these logs for her vote identifier and vote code. If they are present in the log, Alice's vote was cast-as-intended. Alice has to be able to find and open the server logs (A6). If Alice cannot perform this step, she cannot verify her vote. It also is assumed that she can search these logs for her vote identifier and vote code (A7). This means that it is assumed that Alice has previously recorded these values without errors (A2). If Alice did not record these values, she cannot verify. If she made an error during recording, she will not find her vote in the log and therefore assume an error (false positive). If Alice is not able to find her verification code in server logs although it is present, she will also assume an error. Finally, Alice has to be able to compare her vote code without errors (A5).

4 Discussion

In the modified cognitive walkthrough we determined seven assumptions that are made by the approaches and described the impact if these assumptions are violated. In this section we discuss the identified assumptions, the investigated approaches in context of the

assumptions they rely on, and interconnections of security and usability of Internet Voting systems in context of assumptions for cast-as-intended verification.

Assumptions. Assumption A5 (Comparison) is made by all investigated approaches. Therefore, the task of comparing verification codes or other data (e.g. the voter's selection) is of crucial importance for verifying successfully. The verification codes are random character sequences, the voter's selection is ordinary text (e.g. the name of a political party). Hence, there is a difference in the difficulty of comparing. Comparing random character sequences, which is required in Neuchâtel (2015) and Du-Vote, is considered more difficult. The human short time memory is limited to chunks of seven [Mi56, Ta17], therefore, verification codes with more than seven characters are difficult to process for the voters. Hence, the visualization of verification codes should be adjusted (e.g. splitting the codes in chunks [Ta17]), to support an error-free comparison. The assumptions A2 (Storing) and A7 (Search) are similar to the comparison task (A5). If a voter has to store a verification code by writing it down or search for it manually, the same aspects apply.

Approaches. Neuchâtel (2015) requires the least assumptions. The voter has to enter the confirmation code (A1) and make two comparisons (A5) (return code and acknowledgement code). Therefore, from a usability perspective Neuchâtel (2015) is the most promising of the investigated approaches. The scalability of Neuchâtel (2015) and other return code-based approaches, however, appears questionable. One return code per voting option is required and its length depends on the total number of voters. Therefore, elections with many voting options and voters as well as complex elections might be problematic. The investigation and improvement of the scalability of return code-based approaches forms an important task of future work. The Du-Vote approach requires the most assumptions. Among them the access to a trusted hardware token. From a usability perspective Du-Vote is the least promising approach. The assumptions A2 (Storing) and A5 (Comparison) could be removed by providing a device that supports the voter. Therefore, at least A4 would have to be added.

Security and Usability. The usability of the cast-as-intended verification is crucial for the security of the Internet Voting system, but other aspects of security should be considered as well. All investigated approaches rely on different security-related assumptions for ensuring the integrity of the election result. An example is the trustworthiness of the verification devices. Helios, the Estonian System and Du-Vote fail to provide verifiability unless the voter has access to a trustworthy verification device or hardware token (A4). This might be difficult to ensure if the devices in the voter's possession (i.e. a personal computer and a smartphone) are synchronized. Malware infection of one device can also affect the other device (see e.g. [KvdVB16] for elaborating on such vulnerabilities). Neuchâtel (2015) does not require such trusted devices, but requires auxiliary materials. This auxiliary material is a code sheet that is assumed to be generated and distributed in a trustworthy

manner. While the differences in the security models of the investigated approaches are out of scope of this paper, they have to be investigated in conjunction with usability-related assumptions in order to provide a holistic evaluation and comparison of the Internet Voting systems. Another example of the interconnection between the usability of cast-as-intended approaches and the security of the Internet Voting system is the attempt to reduce burden on the voter by automating steps of the verification process. A prominent example is the comparison that the voters have to perform (A5). The comparison could be automated via a supporting software [Ta17]. Automation aims to simplify the verification process and to increase the usability of the cast-as-intended approaches. However, it also introduces new security-related assumptions. Such an assumption would be trust in the supporting software performing the comparison. Such a trade-off between the assumptions should therefore be further investigated in each individual election setting.

5 Related Work

The usability of cast-as-intended approaches in Internet Voting has been investigated in few works so far [WH09, Ka11b, FR12, Ac14, Re17]. Several works [WH09, Ka11b, Ac14, Re17] focus on the usability of the Benaloh Challenge [Be06, Be07] which is used in Helios [Ad08]. The usability of Internet Voting systems that provide return code-based verification was evaluated in [FR12, Ku17]. While these studies provide valuable insights into the usability of the cast-as-intended verification, they focus on evaluating the usability of specific implemented Internet Voting systems. The required voter interaction, however, is given by the process that a voter has to follow in order to verify which is user-interface-independent.

6 Summary and Future Work

In this paper we investigate four approaches for providing for cast-as-intended verifiability. The approaches provide the possibility for the voters to verify that their true intents are represented electronically. Each approach makes assumptions regarding voter capabilities. In our investigation we show that assumptions on voter capabilities have an impact on the Internet Voting system's security and on the integrity of the election result. We identified Neuchâtel (2015) as most promising approach from the usability perspective. Helios and the Estonian system form a middle ground which is followed by the Du-Vote approach. This is a stepping stone into a holistic evaluation and comparison of different cast-as-intended approaches.

Usability is not the only aspect with an impact on the security of the Internet Voting system. There is a tradeoff of usability, security and potentially other influence variables in the scope of cast-as-intended verification. The investigation of this tradeoff forms an important task of future work.

References

[Ac14] Acemyan, C. Z.; Kortum, P.; Byrne, M. D.; Wallach, Dan S.: Usability of Voter Verifiable, End-to-End Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II. In: The USENIX Journal of Election Technology and Systems (JETS), 2(3). USENIX Association, pp. 26–56, 2014.

[Ad06] Adida, B.: Advances in Cryptographic Voting Systems. PhD thesis, Massachusetts Institute of Technology, 2006.

[Ad08] Adida, B.: Helios: Web-based Open-Audit Voting. In: Proceedings of the 17th USENIX Security Symposium. USENIX Association, pp. 335–348, 2008.

[Be06] Benaloh, J.: Simple Verifiable Elections. In: Proceedings of the 1st Electronic Voting Technology Workshop (EVT). USENIX Association, 2006.

[Be07] Benaloh, J.: Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In: In Proceedings of the 2nd Electronic Voting Technology Workshop (EVT). USENIX Association, 2007.

[FR12] Fuglerud, K. S.; Røssvoll, T. H.: An Evaluation of Web-Based Voting Usability and Accessibility. In: Universal Access in the Information Society (UAIS), 11(4). Springer, pp. 359–373, 2012.

[GC16] Guasch Castelló, S.: Individual Verifiability in Electronic Voting. PhD thesis, Universitat Politécnica de Catalunya, 2016.

[GGP15] Galindo, D.; Guasch, S.; Puiggalí, J.: 2015 Neuchâtel's Cast-as-Intended Verification Mechanism. In: Proceedings of the International Conference on E-Voting and Identity (VoteID). Springer, pp. 3–18, 2015.

[Gr15] Grewal, G. S.; Ryan, M. D.; Chen, L.; Clarkson, M. R.: Du-Vote: Remote Electronic Voting with Untrusted Computers. In: Proceedings of the 28th Computer Security Foundations Symposium (CSF). IEEE, pp. 155–169, 2015.

[GV10] Gharadaghy, R.; Volkamer, M.: Verifiability in Electronic Voting-Explanations for Non Security Experts. In: International Conference on Electronic Voting (EVOTE). Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, pp. 151–162, 2010.

[HW14] Heiberg, S.; Willemson, J.: Verifiable Internet Voting in Estonia. In: Proceedings of the International Conference on Electronic Voting: Verifying the Vote (EVOTE). IEEE, pp. 1–8, 2014.

[Ka11a] Karayumak, F.; Kauer, M.; Olembo, M. M.; Volk, T.; Volkamer, M.: User Study of the Improved Helios Voting System Interfaces. In: Proceedings of the 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST). IEEE, pp. 37–44, 2011.

[Ka11b] Karayumak, F.; Olembo, M. M.; Kauer, M.; Volkamer, M.: Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System. In: Proceedings of the Electronic Voting Technology Workshop/ Workshop on Trustworthy Elections (EVT/WOTE). USENIX Association, 2011.

[Ku17] Kulyk, O.; Neumann, S.; Budurushi, J.; Volkamer, M.: Nothing Comes for Free: How Much Usability Can You Sacrifice for Security? In: IEEE Security & Privacy, 15(3). IEEE, pp. 24–29, 2017.

[KvdVB16] Konoth, R. K.; van der Veen, V.; Bos, H.: How Anywhere Computing Just Killed your Phone-Based Two-Factor Authentication. In: Proceedings of the International Conference on Financial Cryptography and Data Security (FC). Springer, pp. 405–421, 2016.

[KW17] Khazaei, S.; Wikström, D.: Return Code Schemes for Electronic Voting Systems. In: Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID). Springer, pp. 198–209, 2017.

[Mi56] Miller, G. A.: The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. In: Psychological Review, 63(2). American Psychological Association, pp. 81–97, 1956.

[Ne14] Neumann, S.; Olembo, M. M.; Renaud, K.; Volkamer, M.: Helios Verification: To Alleviate, or to Nominate: Is That the Question, or Shall we Have Both? In: Proceedings of the International Conference on Electronic Government and the Information Systems Perspective (EGOVIS). Springer, pp. 246–260, 2014.

[Po92] Polson, P. G.; Lewis, C.; Rieman, J.; Wharton, C.: Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. In: International Journal of Man-Machine Studies, 36(5). Elsevier, pp. 741–773, 1992.

[Pu17] Puiggalí, J.; Cucurull, J.; Guasch, S.; Krimmer, R.: Verifiability Experiences in Government Online Voting Systems. In: Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID). Springer, pp. 248–263, 2017.

[Re17] Realpe-Muñoz, P.; Collazos, C. A.; Hurtado, J.; Granollers, T.; Muñoz-Arteaga, J.; Velasco-Medina, J.: Eye Tracking-Based Behavioral Study of Users Using E-Voting Systems. In: Computer Standards & Interfaces. Elsevier, 55, pp. 182–195, 2017.

[Ta17] Tan, J.; Bauer, L.; Bonneau, J.; Cranor, L. F.; Thomas, J.; Ur, B.: Can Unicorns Help Users Compare Crypto Key Fingerprints? In: Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI). ACM, pp. 3787–3798, 2017.

[WH09] Weber, J.-L.; Hengartner, U.: Usability Study of the Open Audit Voting System Helios. http://www.jannaweber.com/wpcontent/uploads/2009/09/858Helios.pdf, 2009. Accessed: 22-December-2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 209

Secure Remote Computation using Intel SGX

David Übler[1], Johannes Götzfried[1], Tilo Müller[1]

Abstract: In this paper, we leverage SGX to provide a secure remote computation framework to be used in a cloud scenario. Our framework consists of two parts, a local part running on the user's machine and a remote part which is executed within the provider's environment. Users can connect and authenticate themselves to the remote side, verify the integrity of a newly spawned loading enclave, and deploy confidential code to the provider's machine. While we are not the first using SGX in a cloud scenario, we provide a full implementation considering all practical pitfalls, e.g., we use Intel's Attestation Services to prove the integrity of the loading enclave to our users. We also take care of establishing a secure bidirectional channel between the target enclave and the client running on the user's machine to send code, commands, and data. The performance overhead of CPU-bound applications using our framework is below 10% compared to remote computation without using SGX.

Keywords: Intel SGX; Cloud Computing; Isolation

1 Introduction

Cloud services are a promising way for companies to lower their expenses for building and maintaining IT infrastructures. A recent survey shows that a majority of companies already shifted some of their workload to cloud providers [ri16] while the market share of cloud services is still growing rapidly. However, providers of these services often cannot be trusted and any promises on confidentiality made by them cannot be enforced by the user.

If a company wants to retain confidentiality and integrity of its code and data, the cloud provider must be prevented from accessing resources while still being able to execute them. This includes, for example, main memory and disk space dedicated to a user's task. In 2013, Intel introduced the Software Guard Extensions (SGX) [Ho13] which are a trusted computing architecture offering isolated memory regions and restricting access to authorized parts of an application. Software attestation can be used to verify those isolated regions by a remote user before secret code and data is transferred to the cloud.

The drawback of using SGX, however, is that a company must explicitly divide its application in trusted and untrusted parts, and set up an attestation scheme. This requires additional implementation effort and imposes a dependency on SGX, as the layout of the program must suit the constraints of SGX. Fulfilling these constraints and correctly designing an application to use SGX quickly becomes nontrivial and leaves ample room for mistakes, and thereby vulnerabilities [G7]. Thus, our approach is to provide an environment to users

[1] Department of Computer Science, FAU Erlangen-Nuremberg, Martensstr. 3, D-91058 Erlangen, david.uebler@posteo.de, {johannes.goetzfried,tilo.mueller}@cs.fau.de

doi:10.18420/sicherheit2018_16

that lets them deploy and execute arbitrary code which can conveniently be protected by the security mechanisms of SGX.

1.1 Our Contribution

Although proposals for SGX-based cloud solutions already exist [BPH14, Sc15], we provide the full implementation of a remote computation framework relying on SGX while considering all practical pitfalls including Intel's Attestation Services. In detail our contributions are:

- Our framework consists of two parts, a local part running on the user's machine and a remote part which is executed within the provider's environment. Users can connect and authenticate themselves to the remote side, verify the integrity of a newly spawned loading enclave, and deploy confidential code at the provider's machine.
- The loading enclave engages in our remote attestation protocol by generating a report containing the enclave's measurement and information needed to establish an encrypted connection to the user. The report is wrapped into a quote with the help of the quoting enclave provided by Intel. This quote is then verified by the user using Intel's Attestation Services.
- If the integrity of the loading enclave can be verified by the user, a bidirectional encrypted connection between the enclave and the user is established to transfer code, commands, and data.
- The performance overhead of CPU-bound applications using our framework is below 10% compared to remote computation without using SGX.

1.2 Related Work

Intel SGX has been mentioned in publications about cloud computing multiple times before. The trusted execution system Haven [BPH14] was designed to securely run unmodified legacy applications, while VC3 [Sc15] offers distributed *Map-Reduce* computations that keep the data being processed hidden from cloud providers. Both solutions, however, did not use real hardware but an SGX emulator provided by Intel, and ignored the complexity of remote attestation in practice. Scone [Ar16] introduced entirely isolated Linux systems by augmenting Docker containers with SGX. Software attestation, however, was also not implemented in Scone. Opaque [Zh17] offers confidentiality for database queries in particular by placing parts of a database within an SGX enclave. So Opaque exclusively aims to secure queries to a SQL database instead of arbitrary programs.

2 Background: Graphene

Graphene [TPV17] is a *library operating system*, that is a library that provides all abstractions and the same environment usually provided by an OS. An application running in this environment can issue system calls and access a filesystem, but the resources exposed to it are not the host's resources. They are rather virtual resources provided by the library OS. As with a virtual machine, all abstractions provided by the library OS do not have to be identical to the abstractions of the host OS. This means that emulating another kernel version, or an entirely different operating system is possible.

Isolation through a library OS can be achieved by reimplementing all system calls and providing a virtual filesystem. System calls do not trap to the host OS, but are rooted in functions within the library OS. Those functions can handle system calls directly, or use the host OS's facilities to handle them. As most software does not issue system calls directly, but uses a standard library, redirecting system calls can usually be done by just modifying the standard library to call the library OS. A filesystem is reached via system calls as well, enabling the library OS to either redirect it to the host, or to use a virtual filesystem.

Graphene was initially designed to protect host systems against malicious guest applications by isolating the guest through virtual system calls. A similar scheme was later added to offer protection against the host to the guest, leveraging SGX.

To protect a guest application, it is entirely executed inside an enclave, including all libraries used and the library OS. This gives the guest's virtual address space the memory protection guarantees provided by SGX. However, a key feature of Graphene is offering the environment of an operating system to allow system calls, file access and dynamic loading. These actions require the guest to communicate with the outside world, receiving results of system calls and data from files controlled by the host. To reduce the risk posed by these outside influences, an additional protection layer is added. The purpose of this layer is to evaluate the host's responses given to system calls and either allow or reject them.

3 Design and Implementation

This section describes the architecture of our framework, the protocol spoken by the local and remote part, as well as selected implementation details.

3.1 Architecture

The overall architecture of our solution follows a client-server model. The client and server can (and should) be executed on different machines connected by a network. The client is responsible for sending code and data to the server, which is loaded into an enclave at runtime on the server side. Once loaded, the client can establish a connection with the deployed code directly.

The parties participating in the described scenario are the *user* and the *provider*. The user is a person, institution, or machine that aims to execute some of its application code using the provider's resources. The provider yields some of its computational resources, such as CPU time, memory and disk space to the user. To this end, the user's machine runs the client code while the provider's machine executes the server code. This distribution splits the whole application into two parts. One part, the user's machine and software, is called the *local side*. The other part, the provider's machine and software is called the *remote side*. Both parts are communicating over a network, which is not considered to be part of either side or in any way secured.

The user has full control over the client software and the machine it is running on. He or she can therefore trust the client side. The remote side, however, is under control of the provider and can not be trusted by the user. From the provider's point of view, the situation is reversed. It can rely on the remote side, being run on its hardware and operating system, but has no guarantee that the user is not malicious. Both parties consider the network being untrusted and make sure not to expose any sensitive information to it.

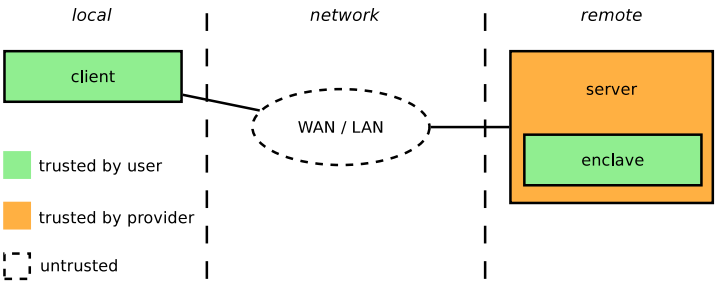


Fig. 1: Remote computation architecture.

While both parties cannot check each others behavior, they can determine the identity of the other side through *attestation*. For example, a guarantee is needed that the provider cannot access the user's code and data after it has been transferred to the remote side. To this end, an SGX *enclave* is running on the provider's machine, alongside the server software. It is at the same time under the user's control and inaccessible to the provider (see Figure 1).

3.2 Protocol

Our protocol has been designed to integrate well into existing networks by building on top of established base protocols. Furthermore, it offers a generic interface to the applications deployed by potential users.

Protocol Stack and State Transitions Our protocol stack consists of different layers. HTTP is used as the base protocol, because it is a broadly accepted protocol and already

used in most existing networks. Using a ubiquitous protocol such as HTTP eliminates the need to reconfigure the network such as opening or forwarding additional ports at routers and firewalls. HTTP messages consist of a header and body, where the body carries the actual payload while the header offers a place to store metadata such as the message's destination and payload length. The body together with the length can be thought of as a *frame*. The layout and contents of those frames is defined by the management layer.

The transitions of the protocol are shown in Figure 2; only client and server take part in the communication. After the connection between client and server has been established, the distributed application is in the *connected* state. In this state, the identities of server and client have not yet been validated and the enclave has not yet been created.

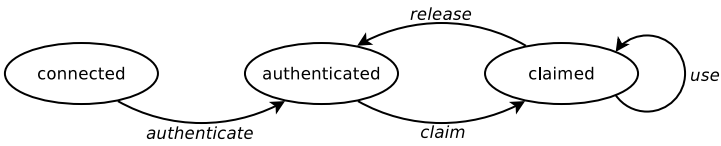


Fig. 2: Protocol states and transitions.

For transition to the next state, the client must authenticate itself to the server. For this transition, a challenge-response protocol based on cryptographic signatures is used. After authentication, the system is in the *authenticated* state. The server now trusts the identity of the client and an encrypted channel is opened between them. Any message arriving on this channel is now guaranteed to have been sent by the client.

When authenticated, the client can request the server to create a new enclave and bind it to the client. This newly created enclave is, however, not usable until it is *claimed*. Claiming consists of the enclave proving its integrity to the client and establishing a second encrypted channel. This second channel is used to deploy the user's code on the enclave. Only after claiming the enclave and sending code, the enclave is within the *claimed* state.

Claiming and Remote Attestation When the server receives a request for a new enclave, it spawns a loading enclave. The loading enclave is not yet bound to a specific client and has an identical layout at the beginning of each claiming process. The initial state has been measured beforehand and is known to both, the client and the server.

Figure 3 shows how SGX remote attestation was adapted to authenticate the loading enclave. In step 1 the client sends a request to the server to launch a new loading enclave. This enclave is spawned in step 2. At this stage the enclave is not yet bound to the client, but to start the binding, the user's public key must be passed to the enclave in step 3.

The public key is part of the secure channel to be established between enclave and client. For this channel to be complete, a keypair and an initial nonce are required as well. The nonce and the keypair is generated in step 4 by the enclave, making the keypair valid only for a

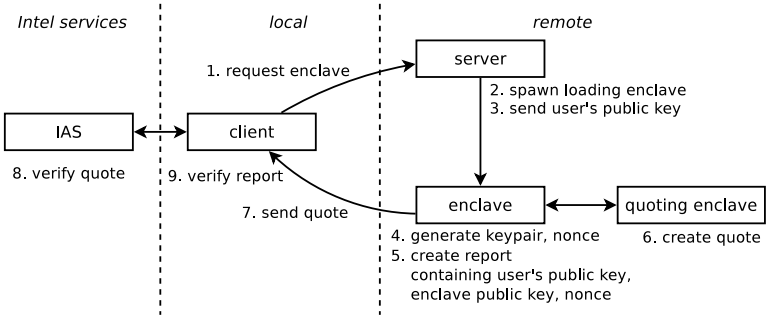


Fig. 3: Remote attestation during enclave claiming.

single session. A permanent keypair is not needed as the enclave's identity is defined by the measurement. Its public and private key have the sole purpose of forming an encrypted channel providing confidentiality, integrity, freshness, and authentication.

The keypair and nonce have to be delivered to the client while at the same time providing proof that the enclave was not tampered with. This proof is given via remote attestation. To this end, the keypair and nonce are embedded in an SGX report, created in step 5. This can be done by copying them to the userdata field of the report. This report is only meaningful on the same platform it was generated. To allow the client to verify the report, a quote is created from it in step 6. This is done by the help of the quoting enclave provided with Intel's SGX SDK.

In step 7, the quote is delivered to the client, which passes it to Intel's Attestation Service, shown in step 8. The attestation service responds to this request with two possible results. If the quote is invalid, the attestation is cancelled and the remote computation session is terminated. Otherwise the client can proceed with step 9, checking the report embedded in the quote. This step can again be done locally without using Intel's services. The report given to the client can now be assumed to be valid. It must, however, be checked if the parameters given in the report are as expected. This is done by comparing the measurement of the enclave to the measurement provided beforehand. Additionally, the user's public key embedded in the report is compared to the one stored locally. By checking the public key, it can be guaranteed that the enclave is bound to the client.

If both values match, the remote attestation and key exchange is complete. A secure channel can be established between both sides using the clients and enclave's keys as well as the nonce. It is important to note that this channel is not accessible to the server. The secret key was computed within the enclave and is thus isolated from the rest of the provider's environment.

Once claimed, the enclave can finally be used. The client is able to send messages to the enclave that are passed to and handled by the user's code. Doing so will cause the system to stay in the *claimed* state. When a user's tasks are complete, and the enclave should no

longer be used, the resources occupied by it will be freed. This is done by issuing a release request to the server. The server will destroy the enclave and the system will fall back to the *authenticated* state. Now, either a new enclave can be requested or the session can be terminated.

3.3 Implementation

We provide an I/O subsystem with basic abstractions to handle all input and output streams used to exchange data between client, server and enclave. These streams are TCP/IP sockets, and on top of a streaming interface, messages are defined as fixed size sequences of bytes, called frames. Our protocol module allows the creation, marshalling and unmarshalling of strongly typed messages to these frames, allowing them to be exchanged using the I/O component.

Encryption is provided on a per-message-parameter basis. Because the server needs partial access to some messages, e.g., the intended recipient of the message, encryption must only be applied to parameters containing sensitive information. To achieve this, the public key authenticated encryption functions provided by the NaCl [Be17] library are used.

An API to be used by client applications and the remote code is provided in C and C++. This allows using the remote computation system with any language that can call C functions, while providing a more comfortable class based interface when using C++.

Graphene as an SGX Environment The enclave is executed in an environment that is isolated from its host in two ways. It has only limited access to the host's resources, by allowing only safe system calls and by having only partial access to the filesystem. To implement these requirements, a sandbox is used to protect the host's system and SGX is used to protect the user's data. With Graphene, a framework is available that provides both. Graphene implements its own virtual filesystem and limits the hosted application to it. It also installs a system call filter that can be configured to allow only certain system calls and with recent versions of Graphene, SGX support is available.

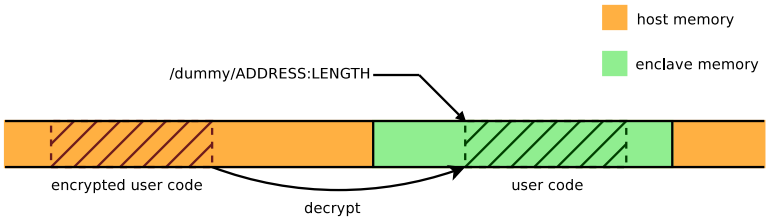


Fig. 4: Pointer filesystem to refer to enclave memory.

Setting up an SGX enclave and performing system calls is handled by Graphene without using SGX functions. Nonetheless, care has to be taken when library calls and system calls

are executed, because data may be copied between enclave and host memory. One such situation arises when code is loaded at runtime. Loading code can be done by using the `dlopen()` function family. However, these functions expect the code to be stored in a file and therefore require the user to pass a filename. There is no way to provide a preallocated buffer directly. Saving the user's code to a file on the provider's disk is unacceptable as well, as it has to be decrypted first to be loaded by `dlopen()`.

To work around this limitation, we use Graphene's virtual filesystem. A new virtual filesystem has been written to map filenames to previously allocated memory regions. A valid filename in this system consists of a memory address, encoded as a decimal number and the length of the memory region in bytes as a decimal number. Using this scheme, the user's code is placed encrypted in the host memory and is decrypted to the enclave's memory. Once it resides in enclave memory, a filename is generated to identify the memory region containing the code. This filename can then be given to `dlopen()`. Figure 4 shows how ciphertext and plaintext reside in host and enclave memory.

4 Performance Evaluation

To evaluate the performance of our secure remote computation framework, it first must be considered which aspects contribute to potential performance losses. The application deployed by the user is inherently distributed, communicating over a network. However, this is not unique to our work when compared to other cloud scenarios and thus, latency and performance loss due to communication, including the marshalling of data, are excluded. Instead we evaluate the overhead of the encryption mechanisms used to protect the user's secrets as well as the sandboxing scheme, including the use of SGX.

CPU and Memory Bound Performance To asses performance of mostly CPU- and memory-bound applications, an example application for edge detection has been developed. The *edgedetect* application reads a PNG file on the local side and sends it to the remote side. The remote side keeps the received PNG in RAM and performs an edge detection algorithm on it. It then sends back the resulting image showing the edges of the original. The resulting image is either displayed or saved on the local side.

Measurements are taken on the local side by starting a timer just before the image is sent over the network. Timing differences includes the transmission, encryption and decryption as well as the algorithm itself. However, it excludes the initial loading of the image from disk, as this is not considered to be part of the remote computation. This measurement cycle was repeated 30 times for four modes, each mode enabling or disabling the sandbox and encryption. In each mode the arithmetic mean, the median and standard deviation was calculated and compared to each other. The ratio was calculated by dividing the median of the mode in question by the median of the fastest mode, the baseline.

Tab. 1: Results for the *edgedetect* testcase.

SGX	encryption	median	s.dev.	ratio to fastest
no	no	84 ms	2 ms	1.00
no	yes	86 ms	3 ms	1.02
yes	no	87 ms	2 ms	1.03
yes	yes	90 ms	4 ms	1.06

Table 1 shows the result of our measurements. Modes running outside the sandbox show shorter time frames. Running the testcase with encryption enabled slows it down by about 2%. Enabling SGX, the testcase takes even longer, but stays within the same order of magnitude with a maximum slowdown of about 6%. During these tests, both memory access and CPU instructions were measured. The encryption only applies to the messages being sent between client and enclave.

System Calls and Disk Usage Our second testcase makes heavy use of I/O operations by writing and reading to files. The *ioslide* testcase sends a configurable number of bytes of arbitrary data from the local to the remote side. The remote side receives the data and enters a loop that is repeated a configurable number of times. Inside the loop, data is simply incremented byte-wise, encrypted and written to a file on the hosts disk as well as to standard output. The file is finally read in again, decrypted and compared to the original data.

This setup gives us two variables to measure: The number of bytes, i.e. the chunk size, determines how many bytes are handled by a system call and the number of iterations that controls how many system calls are executed. In analogy to the *edgedetect* application, the performance of the remote side is measured. The timer is started before sending the byte sequence and stopped after receiving a completion acknowledgement. The time difference includes the transmission, encryption, I/O operations, and decryption of the exchanged data.

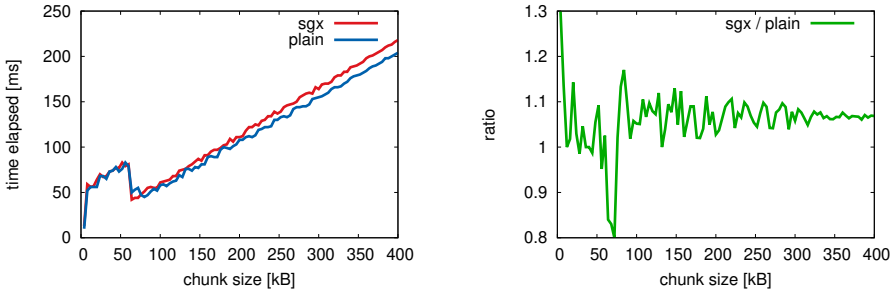


Fig. 5: Results of the *ioslide* testcase with fixed number of iterations but varying chunk size.

Figure 5 shows the result of varying the chunk size but keeping the number of read and write cycles fixed. The time was taken over 100 rounds, each round increasing the chunk size.

The chunk size starts at 4096 bytes and is multiplied by the round number, up to 409,600 bytes or about 400kB. The number of iterations, or cycles, is kept fixed at two. This leads to 8192 bytes being processed in the first round and 819,200 bytes in the last round.

Looking at the left diagram of Figure 5, it can be seen that both the SGX and the plain version remain in close proximity to a common line. Increasing the chunk size increases the time taken to process the data linearly in both versions. This is supported by the right diagram of Figure 5, which shows the ratio between the SGX and plain version. After initial oscillations, it settles on a constant ratio value of about 1.07.

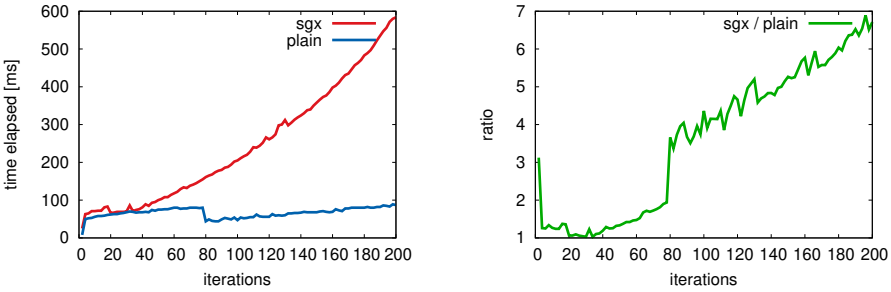


Fig. 6: Running *ioslide* with fixed chunk size and varying number of read/write cycles.

Figure 6 shows the result of keeping the chunk size fixed, but increasing the number of iterations in each round. The number of bytes in each chunk is kept at 4096. The number of cycles starts at two and is multiplied by the round number. Using this configuration, the total number of bytes processed in each round is the same as above, starting at 8192 bytes and leading up to 819,200 bytes.

As seen in the left diagram of Figure 6, the runtime measured of the SGX and plain versions diverge quickly. The time taken by the SGX enabled version increases at a steeper angle than the plain version. This becomes clearer in the right diagram of Figure 6, which shows the time taken by the SGX version divided by the time measured when running the plain version. The ratio settles on a line with a positive slope. It can therefore be concluded that the performance difference between the SGX and plain version increases with each round.

5 Conclusion

We introduced a framework providing remote computation using Intel SGX while considering practical pitfalls such as using Intel's Attestation Services to prove the integrity of the loading enclave to our users. Our framework consists of two parts, a local part running on the user's machine and a remote part which is executed within the provider's environment. Users can connect and authenticate themselves to the remote side, verify the integrity of a newly spawned loading enclave, and deploy confidential code at the provider's machine. The performance overhead is below 10% compared to remote computation without using SGX.

Acknowledgments

This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre “Invasive Computing” (SFB/TR 89).

References

[Ar16] Arnautov, Sergei; Trach, Bohdan; Gregor, Franz; Knauth, Thomas; Martin, Andre; Priebe, Christian; Lind, Joshua; Muthukumaran, Divya; O’Keeffe, Dan; Stillwell, Mark; Goltzsche, David; Eyers, David M.; Kapitza, Rüdiger; Pietzuch, Peter R.; Fetzer, Christof: SCONE: Secure Linux Containers with Intel SGX. In: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI. 2016.

[Be17] Bernstein, Daniel J.: , NaCl: Networking and Cryptography library, July 2017.

[BPH14] Baumann, Andrew; Peinado, Marcus; Hunt, Galen C.: Shielding Applications from an Untrusted Cloud with Haven. In: 11th USENIX Symposium on Operating Systems Design and Implementation, OSDI. 2014.

[G7] Götzfried, Johannes; Eckert, Moritz; Schinzel, Sebastian; Müller, Tilo: Cache Attacks on Intel SGX. In: Proceedings of the 10th European Workshop on Systems Security. EuroSec’17, 2017.

[Ho13] Hoekstra, Matthew; Lal, Reshma; Pappachan, Pradeep; Phegade, Vinay; del Cuvillo, Juan: Using innovative instructions to create trustworthy software solutions. In: Workshop on Hardware and Architectural Support for Security and Privacy. 2013.

[ri16] rightscale: , Cloud Computing Trends: 2016 State of the Cloud Survey, February 2016.

[Sc15] Schuster, Felix; Costa, Manuel; Fournet, Cédric; Gkantsidis, Christos; Peinado, Marcus; Mainar-Ruiz, Gloria; Russinovich, Mark: VC3: Trustworthy Data Analytics in the Cloud Using SGX. In: IEEE Symposium on Security and Privacy. 2015.

[TPV17] Tsai, Chia-che; Porter, Donald E.; Vij, Mona: Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In: USENIX Annual Technical Conference. 2017.

[Zh17] Zheng, Wenting; Dave, Ankur; Beekman, Jethro G.; Popa, Raluca Ada; Gonzalez, Joseph E.; Stoica, Ion: Opaque: An Oblivious and Encrypted Distributed Analytics Platform. In: 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI. 2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 221

Homomorphe Verschlüsselung für Cloud-Datenbanken: Übersicht und Anforderungsanalyse

Lena Wiese[1], Daniel Homann[1], Tim Waage[1], Michael Brenner[2]

Abstract:

Die Auslagerung von Daten in Cloud-Datenbanken verspricht eine Reihe von Vorteilen wie reduzierte Wartungskosten, Flexibilität der Ressourcenverteilung und einfachen, ortsunabhängigen Zugriff. Diese Datenbanken bieten dabei eine Vielzahl von Funktionalitäten, um Berechnungen auf Daten auszuführen. Datensicherheit (einschließlich dem Schutz persönlicher Daten) ist in Cloud-Datenbanken jedoch noch nicht angemessen umgesetzt worden. Konventionelle Verschlüsselungsverfahren garantieren zwar hohe Sicherheit während Transport und Speicherung, verhindern aber auch weitere Berechnungen auf den Daten. Modernere homomorphe Verschlüsselungsverfahren versprechen dagegen sowohl Datensicherheit als auch die Möglichkeit, auf verschlüsselten Daten zu rechnen. Das bestehende System *FamilyGuard* kombiniert bisher eigenschaftsbewahrende Verschlüsselungsverfahren. Um die Funktionalität auf Aggregationsfunktionen zu erweitern, soll in Zukunft auch homomorphe Verschlüsselung eingesetzt werden. In diesem Artikel geben wir eine Übersicht über die verschiedenen Arten homomorpher Verschlüsselungsverfahren und ihre Sicherheitsgrundlagen. Im Anschluss stellen wir Anforderungen für den Einsatz homomorpher Verfahren in Cloud-Datenbanken auf.

Keywords: Homomorphe Verschlüsselung; Datenbanken-Sicherheit

1 Einleitung

Moderne Clouddienste ermöglichen das effiziente Verwalten großer Datensätze. Mit solchen Diensten kann zudem zur Datenanalyse moderne Hochperformanz-Hardware als Platform- oder Software-as-a-Service genutzt werden. Datenbanken können zum Einen Daten zuverlässig speichern und zum Anderen Berechnungen auf den gespeicherten Daten ausführen. Auch Databases-as-a-Service werden von Dienstanbietern in der Cloud angeboten. Jedoch ist über die Vertrauenswürdigkeit der Dienstanbieter oft kein abschließendes Urteil möglich. So ist es teilweise unklar, in welchem Land die Daten gespeichert werden. Der Einsatz von Datenverschlüsselung wird daher aus Datenschutzgründen notwendig.
Es existieren starke kryptographische Verfahren, um den Schutz vertraulicher Daten beweisbar sicherzustellen. Eine naive Lösung des Datenschutzproblems wäre es also,

[1] Universität Göttingen, Institut für Informatik, Goldschmidtstraße 7, 37077 Göttingen, Germany, E-Mail: {wiese, homann, waage}@cs.uni-goettingen.de
[2] Universität Hannover, Fachgebiet Computational Health Informatics, 30159 Hannover, Germany, E-Mail: brenner@luis.uni-hannover.de

doi:10.18420/sicherheit2018_17

die Daten mit einem konventionellen symmetrischen Verschlüsselungsverfahren (wie AES) zu verschlüsseln. Ein erheblicher Nachteil ist jedoch, dass bei einem Lesezugriff zwecks Entschlüsselung alle Daten wieder zum Datenbesitzer transferiert werden müssen. Schreibzugriffe führen darüber hinaus zu weiteren Verschlüsselungsschritten und Zugriffe durch mehrere Besitzer erfordern zusätzlichen Aufwand für eine Schlüsselverwaltung. Neben dem Vertraulichkeitsschutz gilt es daher auch die Daten effizient verarbeiten zu können. Eine Option ist die Entwicklung spezieller kryptographischer Verfahren, mit denen verschlüsselte Daten ausgewertet werden können.
Ein konkreter Anwendungsfall ist die Analyse medizinischer Datensätze in der Cloud (zum Beispiel in der personalisierten Medizin). Solche Datensätze und Clouddienste existieren bereits (wie zum Beispiel unter `http://www.biobanken.de`) und ihre Zahl wächst stetig – besonders für personalisierte Diagnosen oder Behandlungen [Sh17]. Die Vertraulichkeit personenbezogener Daten ist in medizinischen Anwendungen extrem wichtig – etwa bei der Ausführung medizinischer Analysen (zum Beispiel die Analyse von Kohorten ähnlicher Patienten). Die Auswertung von medizinischen Daten in Clouddiensten erfordert jedoch neue Sicherheitsverfahren, um den Schutz von Patientendaten zu gewährleisten.

2 Hintergrund

2.1 Verschlüsselungsverfahren

Der Schutz der Daten und ihre Nutzbarkeit sind in der Regel zwei gegensätzliche Anforderungen. Kommt es zur Verarbeitung vertraulicher Daten, sollten eine vollständige Entschlüsselung, Verarbeitung und anschließende Neuverschlüsselung vermieden werden. Stattdessen sollte eine (Vor-)Verarbeitung der verschlüsselten Daten möglich sein, um die gewünschten Analyseergebnisse unter Aufrechterhaltung der Vertraulichkeit zu erhalten. Dies wird durch spezielle Verschlüsselungsverfahren erreicht, mit denen zumindest Teile der Daten verschlüsselt werden, bevor sie in einem Clouddienst gespeichert werden:

- **eigenschaftsbewahrende** Verschlüsselungsverfahren (engl.: property-preserving encryption; PPE), bei denen sich Eigenschaften des Klartextes auf den verschlüsselten Text übertragen. Hier existieren Verfahren zur durchsuchbaren und ordnungsbewahrenden Verschlüsselung, so dass eine Suche nach verschlüsselten Suchworten und eine Sortierung verschlüsselter Daten möglich ist.
- **homomorphe** Verschlüsselungsverfahren (engl.: homomorphic encryption; HE), mit denen eine Auswertung auf verschlüsselten Daten zu einem verschlüsselten Ergebnis führt, das der anfragende Benutzer dann erhält und entschlüsselt. Der Clouddienst kann also weder die gespeicherten Daten noch das Ergebnis selbst entschlüsseln.

2.2 Einsatz von Verschlüsselung in Datenbanken

Im Projekt FamilyGuard wurden erstmalig mehrere durchsuchbare und ordnungsbewahrende Verschlüsselungsverfahren in einem einheitlichen Rahmen implementiert, vergleichend mit den Spaltenfamilien-Datenbanken HBase und Cassandra getestet und ihre Sicherheitseigenschaften analysiert [WW17]. Spaltenfamilien-Datenbanken gehören zur Gruppe der nichtrelationalen („NoSQL“) Datenbanken [Wi15]. Sie werden wegen ihres flexiblen und effizienten Verhaltens gerne von Cloudspeicher-Anbietern verwendet, bei denen verschiedene Kunden Speicherplatz mieten können. Dies führt zu Bedenken, wie vertrauliche Daten vor einem neugierigen Cloudspeicher-Anbieter oder anderen Angriffen von unbekannten Dritten geschützt werden können. Bisher bieten Spaltenfamilien-Datenbanken keine Methoden an, mit denen der Schutz vertraulicher Daten vor unbefugtem Zugriff sichergestellt werden kann. Das Hauptziel des FamilyGuard-Projektes war es daher, eine sichere und durch den Benutzer anpassbare verschlüsselte Speicherung von Daten in Spaltenfamilien-Datenbanken zu ermöglichen. Dieses Ziel wurde im Projekt FamilyGuard erreicht, indem Verfahren der durchsuchbaren und ordnungsbewahrenden Verschlüsselung für Spaltenfamilien-Datenbanken nutzbar gemacht wurden. Mit diesen (nun erstmals in Java quelloffen verfügbaren) Verfahren können die Datenbanken die folgenden zwei Funktionalitäten anbieten:

1. verschlüsselte Daten können nach verschlüsselten Suchworten durchsucht werden;
2. verschlüsselte numerische Daten können sortiert werden und damit auch Bereichsabfragen (nach Intervallen von Werten) beantwortet werden.

Aus den existierenden Verfahren zur durchsuchbaren und ordnungsbewahrenden Verschlüsselung konnten solche Schemata identifiziert und implementiert werden, die den Anforderungen an den praktischen Einsatz in den unmodifizierten Datenbanksystemen entsprechen und die jeweils bestmöglichen Sicherheitseigenschaften in ihrer Kategorie aufweisen. Eine umfassende Implementierung verschiedener geeigneter Verfahren steht bereits als quelloffenes Framework unter `https://github.com/dbsec/FamilyGuard/` zur Verfügung. Die Performanz konnte sowohl mit synthetischen Benchmarks als auch mit praxisorientierten Messungen quantifiziert werden. In zukünftigen Arbeiten sollen homomorphe Verschlüsselungsverfahren in das bestehende System eingebunden und mit den bereits vorhandenen Verfahren (also durchsuchbaren und ordnungsbewahrenden Verschlüsselungsverfahren) kombiniert werden. Unsere Vision ist eine „verschlüsselte Datenspeicherung als Dienst“ für die Spaltenfamilien-Datenbanken. Auf diese Weise kann der Kunde dann mit der Cloud-Datenbank vielseitig interagieren:

- zum Einen kann er Berechnungen auf den gespeicherten Daten ausführen
- zum Anderen lassen sich diese Berechnungen (auf Grundlage der homomorphen Verschlüsselung) kombinieren mit Selektionsanfragen (auf Grundlage der bereits vorhandenen durchsuchbaren und ordnungsbewahrenden Verschlüsselung)

Andere Systeme bieten diese Funktionalität nur eingeschränkt an. Als eine alternative Implementierung, die sowohl homomorphe als auch eigenschaftsbewahrende Verfahren in Cloud-Datenbanken einsetzt, bietet das Projekt CryptDB [Po12; PZB15] eine sogenannte SQL-bewusste Verschlüsselung. An homomorphen Verfahren werden aber nur die beiden einfachsten Verfahren zu rein additiv-homomorpher Verschlüsselung (Paillier [Pa99]) und zu rein multiplikativ-homomorpher Verschlüsselung (ElGamal [Ga84]) implementiert. Das bedeutet, dass nur eingeschränkte Funktionen (nur Addition oder nur Multiplikation) auf den verschlüsselten Daten berechnet werden können; es können so keine generellen Funktionen berechnet werden, die eine Kombination von Addition und Multiplikation benötigen. Außerdem kann kein symmetrisches Entschlüsselungsverfahren berechnet werden, wie es für die delegierte Verschlüsselung (siehe Abschnitt 3.4) nötig ist. Monomi [Tu13] erweitert CryptDB um mehr SQL-Anfragen, die nun ausgeführt werden können – jedoch mit höherem Rechenaufwand und Speicherplatzbedarf auf Kundenseite (statt auf Cloudseite).
Weitere Ansätze verlangen teure kryptographische Hardware auf Seiten der Cloud-Datenbank (wie etwa TrustedDB [BS14] oder Cipherbase [Ar13]). Es findet dann eine Entschlüsselung der Daten auf Seiten der Cloud-Datenbank innerhalb der kryptographischen Hardware statt und der Cloudanbieter benötigt dazu immer den geheimen Schlüssel des Benutzers; daher sind diese Hardware-basierten Ansätze für eine datenschutzkonforme Analyse ungeeignet.

2.3 Einsatz von Verschlüsselung zur Programmausführung

Neben der etablierten Transportverschlüsselung sollen durch den Einsatz von Verschlüsselung während der Programmausführung auch Sicherheitsmechanismen während der aktiven Datenverarbeitung auf nichtvertrauenswürdigen Ressourcen ermöglicht werden. Dazu werden zur Zeit sowohl einzelne Techniken des verschlüsselten Rechnens, wie homomorphe Kryptographie, Secure Multiparty Computation oder Secure Function Evaluation.
Im Projekt *hcrypt* (`http://hcrypt.com`) ist ein verschlüsseltes Prozessormodell entstanden, mit dem beliebige, verschlüsselte, nicht-sequenzielle Programme mit freiem Lese- und Schreibzugriff auf verschlüsselten Speicher ausgeführt werden können [PBS11]. Diese generische Rechenmaschine kann zwar alle möglichen Programme ausführen, sie unterliegt aufgrund der notwendigen sequentiellen Abarbeitung der rein homomorph verschlüsselten Schaltkreisgatter erheblichen Laufzeitbeschränkungen.
Während dieses Modell die Wirksamkeit der Verschlüsselung zur Laufzeit belegt, offenbart es zugleich die Notwendigkeit der Effizienzsteigerung, um praktische Anwendbarkeit zu erlangen. Dies kann grundsätzlich auf zwei verschiedene Arten erfolgen: Zum Einen basiert die Performanz des Modells auf der Leistungsfähigkeit der Implementierung der verwendeten homomorphen Verschlüsselung. Das heißt, dass sich Fortschritte in diesem Bereich automatisch auf die Effizienz der Anwendungen auswirken. Zum Anderen kann die Effizienz durch den Einsatz geschickt arrangierter Protokolle wesentlich verbessert werden, bei denen konstruktive Maßnahmen ergriffen werden [BS13] oder bei denen nicht die gesamte Verarbeitung auf homomorpher Kryptografie beruht und hybrid mit anderen Technologien verbunden wird.

3 Homomorphe Verfahren

Wir geben nun eine Übersicht über Varianten der homomorphen Verschlüsselung. Tabelle 1 listet die der Sicherheit der Verfahren zugrundeliegenden mathematischen Probleme auf.

3.1 Additiv-homomorphe Verschlüsselung

Eine additiv-homomorphe Verschlüsselungsfunktion *Enc* hat die Eigenschaft, dass sich zu zwei Schlüsseltexten *Enc(a)* und *Enc(b)* effizient die verschlüsselte Summe *Enc(a + b)* berechnen lässt. Das grundlegende Verfahren zur additiv-homomorphen Verschlüsselung ist das Paillier-Verfahren [Pa99], das auf der Zusammengesetze-Reste-Annahme basiert. Es gibt zudem folgende neuere additiv-homomorphe Verfahren:

- Das Joye-Libert-Verfahren [JL13] basiert auf dem traditionellen Goldwasser-Micali-Verfahren [GM84], hat aber eine höhere Effizienz bei der Entschlüsselung. Das Verfahren ist sicher unter der Quadratische-Reste-Annahme.
- Das Castagnos-Laguillaumie-Verfahren [CL15] ist sicher unter der Diffie-Hellman-Annahme.
- Das Fousse-Lafourcade-Alnumaimi-Verfahren [FLA11] ist sicher unter der Höhere-Reste-Annahme.

3.2 Multiplikativ-homomorphe Verschlüsselung

Eine multiplikativ-homomorphe Verschlüsselungsfunktion hat die Eigenschaft, dass sich zu zwei Schlüsseltexten *Enc(a)* und *Enc(b)* effizient das verschlüsselte Produkt *Enc(a × b)* berechnen lässt. Das grundlegende Verfahren zur multiplikativ-homomorphen Verschlüsselung ist das ElGamal-Verfahren [Ga84] und seine auf elliptischen Kurven basierende Variante [Ko87]. Zudem gibt es als neueres rein multiplikativ-homomorphes Verfahren das Desmedt-Gennaro-Kurosawa-Shoup-Verfahren [De10]: es ist sicher unter der Diffie-Hellman-Annahme und baut auf dem Cramer-Shoup-Verfahren [CS98] auf.

3.3 Vollhomomorphe Verschlüsselung

Die Idee beliebiger Kombinationen von Additionen und Multiplikationen auf verschlüsselten Daten wurde bereits 1978 von Rivest, Adleman und Dertouzos beschrieben [RAD78]. Es gab jedoch lange kein Verfahren (auch keine theoretische Konstruktion) eines solchen vollhomomorphen Verfahrens. Aus diesem Grund und aus Effizienzgründen wurden begrenzt („somewhat“) homomorphe Verfahren entwickelt. In diesen Verfahren ist die Anzahl der

Additionen in der Regel unbeschränkt jedoch die Anzahl der Multiplikationen begrenzt. Erst im Jahre 2009 wurde von Craig Gentry ein theoretisches vollhomorphes Verschlüsselungsverfahren vorgeschlagen. Weil nun die logischen Operatoren XOR und AND auf verschlüsselten Daten ausgeführt werden können, ist es möglich beliebige Operationen beliebig oft auf den verschlüsselten Daten zu berechnen. Die Sicherheit von Gentry's erster Konstruktion [Ge09; Ge10] eines vollhomomorphen Verfahrens basiert auf der Komplexität zweier verschiedener Probleme. Zum einen basiert die Sicherheit der von ihm vorgeschlagenen homomorphen Verschlüsselung auf der Komplexität des Shortest Independent Vector Problems (SIVP) in von Idealen erzeugten Gittern. Zum anderen basiert die Sicherheit des von Gentry entworfenen Bootstrapping-Algorithmus auf der Komplexität des Sparse Subset Sum Problems. Grundsätzlich liegt ein wesentliches Problem der homomorphen Multiplikation darin, dass jeder Multiplikationschritt das Ergebnis verfälscht: es treten Ungenauigkeiten durch Rauschen (engl.: noise) im Berechnungsergebnis auf, die sich mit jeder Multiplikation verstärken. Dadurch ist die Anzahl der ausführbaren Multiplikationen begrenzt, wenn man ein korrektes Ergebnis erhalten will. Diese Ungenauigkeit des Ergebnisses wird durch das Gentry-Verfahren vermieden, indem ein Auffrischen der verschlüsselten Daten nach einer gewissen Anzahl von Multiplikationen durchgeführt wird; dieses Auffrischen nennt Gentry „bootstrapping". Da dieses Auffrischen sehr zeitaufwändig ist, wurden zahlreiche Optimierungen [Di10] vorgeschlagen.
Aktuell können mit der sogenannten nivellierten homomorphen Verschlüsselung (engl.: leveled homomorphic encryption) die Parameter so optimiert werden, dass eine gewählte Anzahl von Multiplikationen ausgeführt werden kann, ohne dass die Ungenauigkeiten zu groß werden [BGV14]. Derzeit gilt das Fan-Vercauteren-Verfahren [FV12] als das effizienteste bei den höchsten Sicherheitseigenschaften. Die Sicherheit dieses Algorithmus basiert auf dem Ring Learning With Errors (RLWE) Problem; es optimiert das Verfahren von Brakerski [Br12], das auf dem Learning with Errors (LWE) Problem basiert.
Ein weiteres nivelliertes Verfahren ist das YASHE-Schema [Bo13]. Sein Sicherheitsbeweis basiert auf der NTRU-Annahme [HPS98], die besagt, dass es schwer ist in einem speziellen Verband einen kürzesten Vektor zu finden, beziehungsweise zu einem nicht am Verband beteiligten Vektor den nächstliegenden Vektor im Verband zu finden. Jedoch basiert das Schema auf einer überdehnten (engl.: „overstretched") Variante der NTRU-Annahme, für die bereits verschiedene Angriffe entdeckt wurden [ABD16; CJL16; KF16].
Zahlreiche Verfahren bauen auf diesen grundlegenden Schemata auf, um weitergehende Funktionalitäten zu ermöglichen – so zum Beispiel auch Mehrbenutzerverfahren, die verschiedene Benutzer-Schlüssel unterstützen [LTV17].

3.4 Delegierte Verschlüsselung

Um den Nutzer von der langwierigen vollhomomorphen Verschlüsselung zu entlasten, wurde ein Verfahren vorgeschlagen, um die homomorphe Verschlüsselung an einen Cloudanbieter zu delegieren ohne Klartextdaten preiszugeben. Der Kunde muss nur eine schnelle symmetrische Verschlüsselung (z.B. AES) auf seinen Daten ausführen und die so verschlüsselten

Name (Englisch)	Beschreibung
Shortest Independent Vector	Gegeben ein n-dimensionales Gitter L und eine Konstante $\gamma > 1$. Bestimme linear unabhängige Vektoren $v_1, \ldots v_n \in L$, sodass $\max \|v_i\| < \gamma\lambda(n)$. Dabei ist $\lambda(n)$ die minimale Norm von n unabhängigen Vektoren aus L.
Sparse Subset Sum	Bestimme $A' \subset A$, sodass $\sum_{a \in A'} a \equiv t \mod M$ für $A \subset \mathbb{N}$ und $t, M \in \mathbb{N}$.
Learning With Errors	Bestimme eine lineare Funktion $f : \mathbb{Z}_q^n \to \mathbb{Z}_q$, sodass $y = f(x)$ mit hoher Wahrscheinlichkeit für Stichproben (x, y) mit $x \in \mathbb{Z}_q^n$ und $y \in \mathbb{Z}_q$ gilt.

Tab. 1: Den homomorphen Verschlüsselungs-Verfahren zugrundeliegende mathematische Probleme

Daten dann in der Cloud-Datenbank speichern. Nur der kurze symmetrische Schlüssel muss auf Kundenseite dann noch homomorph (mit dem öffentlichen Schlüssel des Kunden) verschlüsselt und dem Clouddienst zur Verfügung gestellt werden. Der Clouddienst verschlüsselt dann die Daten ein zweites Mal homomorph (ebenfalls mit dem öffentlichen Schlüssel des Kunden) und führt danach mittels des homomorph verschlüsselten symmetrischen Schlüssels eine Entschlüsselung des symmetrischen Verfahrens durch. Danach liegen die Daten dann allein homomorph verschlüsselt vor, womit die zeitaufwändige homomorphe Verschlüsselung an den Clouddienst delegiert wurde. Bisher entwickelte homomorphe Verfahren arbeiten in Kombination mit AES oder dem leichtgewichtigen SIMON [Ca15; CLT14; GHS12; LN14; NLV11].

Diese Delegation der homomorphen Verschlüsselung ist wichtig für ressourcenarme Endgeräte wie sie zum Beispiel zunehmend für mobile Gesundheitsanwendungen eingesetzt werden. Eine Schnittstelle für AES-verschlüsselte Daten bietet auch generell den Vorteil, das auf Benutzerseite nur konventionelle AES-Verschlüsselung benötigt wird; es müssen dort also keine speziellen Verschlüsselungsmodule installiert werden.

Ein Anwendungsfall für die delegierte Verschlüsselung sind Fitnesstracker: diese Fitnesstracker zeichnen zumeist als Armband kontinuierlich die sportliche Betätigung des Nutzers auf und übermitteln sie an eine Smartphone-App; Smartphones selbst sind nicht performant genug, um eine volle homomorphe Verschlüsselung auf großen Mengen von Messdaten auszuführen. Fitness-Daten können aber mit einem Armband generiert werden und zunächst mittels einer Smartpone-App schneller symmetrisch (etwa mit AES) verschlüsselt werden; anschließend können sie dann auf Seiten der Cloud-Datenbank homomorph verschlüsselt werden (durch delegierte Verschlüsselung) und danach Statistiken (zum Beispiel Wochen- oder Tagesdurchschnitte) in der Datenbank berechnet werden.

3.5 Implementierungen homomorpher Kryptosysteme

Seit der Entdeckung des Verfahrens zur Konstruktion unbegrenzt homomorpher ("fully homomorphic") Kryptosysteme aus der Kombination begrenzter homomorpher ("somew-

hat homomorphic") Kryptosysteme und einem sogenannten Bootstrapping-Mechanismus wurden zunächst Gitter- und Ganzzahl-basierten Verfahren implementiert. Beispiele sind

- M. Brenner, H. Perl, M. Smith (2011): Libscarab; hcrypt-Project `http://hcrypt.com`
- C. Gentry, S. Halevi (2011): Implementing Gentry's Fully Homomorphic Encryption Scheme, Springer LNCS Vol. 6632

Nach einer mehrjährigen Konsolidierungsphase haben sich die RLWE-basierten Systeme durchgesetzt. Bisher wurden Implementierungen unterschiedlichen Umfangs vorgelegt:

- SEAL `http://sealcrypto.org` (Microsoft Research)
- HElib `https://github.com/shaih/HElib` (Shai Halevi / IBM, Victor Shoup)
- NFLlib `https://github.com/CryptoExperts/FV-NFLlib` (Paillier / CryptoExperts)
- Palisade `https://git.njit.edu/groups/palisade` (NJIT)
- cuHE `https://github.com/vernamlab/cuHE` (Worcester Polytechnic Institute)
- HEAAN `https://github.com/kimandrik/HEAAN` (Seoul National University, UCSD)
- TFHE `https://github.com/tfhe/tfhe` (Ilaria Chillotti, Nicolas Gama et al.)

Die tatsächliche Anwendung der genannten Implementierungen unterliegt zur Zeit aufgrund des experimentellen Charakters wesentlichen praktischen Beschränkungen. Insbesondere ist die Interoperabilität derzeit nicht gegeben. Aktuell arbeiten die Gruppen SEAL, HElib, NFLlib, Palisade cuHE, HEAAN und Libscarab zusammen mit Industriepartnern, Universitäten und den US-amerikanischen Behörden NIST und NIH an der Standardisierung homomorpher Verschlüsselungsverfahren (`http://HomomorphicEncryption.org`). Dies umfasst auch eine einheitliche Programmierschnittstelle (API) [Br], um eine Modularität der Programmentwicklung und Austauschbarkeit der Algorithmen zu ermöglichen.

4 Funktionale Anforderungen für den Einsatz homomorpher Verfahren in Cloud-Datenbanken

Obwohl die vollhomomorphen Verschlüsselungsverfahren in der Theorie immer weiter verbessert werden, sind sie in der Praxis immer noch schwer anzuwenden aufgrund der exponentiellen Vergrößerung des Schlüsseltextes gegenüber dem Klartext und der langen Berechnungsdauer. Ein weiteres Problem ist die Wahl von geeigneten Parametern, mit denen die Verfahren sicher genug aber gleichzeitig noch effizient in der Praxis sind. Bisherige Implementierungen sind wenig praxistauglich aus folgenden Gründen:

1. Die Implementierungen laufen nur in unrealistischen Umgebungen ohne Cloudanbindung (insbesondere keine Cloud-Datenbanken als Dienst). Beispielsweise wird die oben beschriebene delegierte Verschlüsselung zwar in [GHS12] und [CLT14] für AES und in [LN14] für ein anderes symmetrische Verfahren (namens SIMON) getestet jedoch nicht innerhalb einer realistischen Cloudumgebung. Nach der delegierten Verschlüsselung werden bei den bisherigen Tests auch keine weiteren Operationen auf den verschlüsselten Daten ausgeführt. Es fehlt daher bisher der entscheidende Nachweis, ob dieses Verfahren der delegierten Verschlüsselung in der Praxis so funktioniert, dass die Daten noch weiterverarbeitet werden können.
2. Die Implementierungen sind optimiert für eingeschränkte Spezialanwendungen (wie etwa Assoziationsstudien auf verschlüsselten Genom-Daten [KL15]). Wünschenswert wäre jedoch ein Verschlüsselungsdienst, der vielseitiger einsetzbar ist.
3. Die Implementierungen vergleichen nicht mehrere Verfahren und bieten daher keinen Anhaltspunkt, welches Verfahren am besten geeignet ist. Die einzige Ausnahmen sind hier [LN14], die zwei homomorphe Verfahren vergleichen und [CS16], die vier homomorphe Verfahren vergleichen; jedoch wurde die Implementierungen nicht in Cloudumgebungen oder einem Datenbanksystem getestet.

Für einen realistischen Einsatz von homomorphen Verschlüsselungsverfahren in Cloud-Datenbanken ergeben sich zahlreiche funktionale Anforderungen:

Nichtinteraktivität: Mehrere Kommunikationsrunden (zwischen Server und Kunden) sind für eine Datenbankanwendung zu ineffizient. Daher sollte idealerweise eine Datenabfrage jeweils nur eine einmalige Kommunikation zwischen Datenbank und Nutzer erfordern. Die homomorphen Verfahren sind grundsätzlich für diese Funktionalität geeignet. Diese Anforderung kann gegebenenfalls abgeschwächt werden: zu Gunsten einer besseren Performanz können homomorphen Verfahren mit Mehrparteien-Berechnungen kombiniert werden [LTV17].

System-Unabhängigkeit: Die bestehenden Cloud-Datenbanken sollen in ihrer Funktionsweise möglichst wenig verändert werden. Die Weiterentwicklung der Datenbanksysteme soll unabhängig von den Sicherheitsfunktionalitäten erfolgen können. Zugleich soll die Umsetzung der homomorphen Addition und Multiplikation innerhalb des Datenbanksystems erfolgen. Idealerweise sollen die Berechnungen auf verschlüsselten Daten daher durch benutzerdefinierte Funktionen in der Datenbank umgesetzt werden, die die Funktionalität des Datenbanksystems erweitern.

Modularität: Die Implementierung soll sich nicht auf ein spezielles Verschlüsselungsverfahren festlegen, sondern es sollen sich verschiedene Verschlüsselungsalgorithmen einbinden lassen. Damit wird sichergestellt, dass aktuelle Entwicklungen im Bereich der homomorphen Verschlüsselung direkt genutzt werden können.

Konfigurierbarkeit: Bei den einzelnen homomorphen Verfahren müssen konkrete Parameter gefunden werden, die eine gute Abwägung zwischen Effizienz und Sicherheit darstellen [CS16]; solche Parameter sind zum Beispiel die Schlüssellänge, der Modulus, die Länge der zu verschlüsselnden Worte oder die Anzahl der Multiplikationen. Diese Abwägung sollte die Dauer des Schutzbedarfs einschließen: Transaktionsdaten verfallen oft direkt nach der Transaktion, langzeitgespeicherte Daten brauchen stärkere Schlüssel. Dies wird auch in einem Bericht der ENISA berücksichtigt [Sm14].

Schnittstelle mit delegierter Verschlüsselung: Für die Umsetzung der delegierten Verschlüsselung muss sowohl ein geeignetes homomorphes Verfahren gefunden werden, das auf dem Clouddienst ausgeführt wird, als auch ein symmetrisches Verfahren, das auf Nutzerseite ausgeführt wird. Zusätzlich müssen Verschlüsselungs- und Entschlüsselungsmethoden des symmetrischen Verfahrens homomorph innerhalb des Datenbankservers implementiert werden können. Bei der Implementierung der delegierten Verschlüsselung ist die Wahl der Parameter nicht nur für die Sicherheit entscheidend, sondern auch für die Funktionalität: Da die Entschlüsselungsfunktion des symmetrischen Verfahrens auf Seiten der Cloud-Datenbank homomorph ausgeführt werden muss, muss das homomorphe Verfahren eine ausreichende Anzahl von Multiplikationen ausführen können.

5 Sicherheitsanforderungen

Die Sicherheit der homomorphen Verfahren beruht auf der Komplexität verschiedener theoretischer Probleme wie oben beschrieben. Jedoch werden hier oft praktische Aspekte außer Acht gelassen. Ein wichtiger Aspekt insbesondere für Datenbankanwendungen ist es, dass ein Angreifer adaptiv sein kann: aus einer Anfrage- und Antworthistorie für einen Benutzer kann er Informationen ansammeln und damit Kenntnisse über gespeicherte Daten aber auch über neu hinzugefügte Daten gewinnen. Für eigenschaftsbewahrende Kryptoverfahren wurde dieses Problem bereits erkannt [Ke15; NKW15]. Diese Art von Angriffen für homomorphe Verfahren wurden bisher nicht umfangreich untersucht.
Um einen vollen Funktionsumfang zu erreichen, muss ein Attribut gegebenenfalls mit mehreren Verfahren verschlüsselt werden; zum Beispiel sowohl ordnungsbewahrend (um sortieren zu können), durchsuchbar (um nach Suchwörtern filtern zu können) als auch homomorph (um Funktionen berechnen zu können). Dadurch liegen zu einem Klartext mehrere Schlüsseltexte und mehrere Metadaten (zum Beispiel Indexe oder Wörterbücher für ordnungsbewahrende Verschlüsselung) vor. Möglicherweise ergibt sich durch Kenntnis verschiedener Schlüsseltexte und Metadaten des gleichen Klartextes ein Informationsgewinn gegenüber der Kenntnis nur eines Schlüsseltextes. Eine offene Frage ist also, ob die einzelnen Verfahren ihre Sicherheitseigenschaften überhaupt aufrecht erhalten können, wenn mehrere Verschlüsselungen gleichzeitig verwendet werden; dies insbesondere auch in dem Fall, dass der Angreifer sich adaptiv verhält – also die Anfrage-/Antworthistorie eines Benutzers miteinbezieht.

Bei der Sicherheitsanalyse der kryptographischen Algorithmen gehen wir von einem Angreifer aus, der ehrlich aber neugierig (engl.: honest but curious) ist. Dabei beobachtet der Datenbankserver den Inhalt der verschlüsselten Datenbank, die Anfragen und Ergebnisse und versucht, daraus auf den Klartextinhalt zu schließen. Es gibt auch böswillige Angreifer, die den Datenbankinhalt oder Anfrageergebnisse aktiv manipulieren. Jedoch ist jedes homomorphe Verfahren formbar (engl.: malleable): eine Berechnung auf verschlüsselten Eingaben ergibt ein sinnvolles entschlüsselbares Ergebnis. Daher muss dem Datenbankserver in der Regel vertraut werden, dass er die Berechnungen auf den verschlüsselten Eingaben korrekt durchführt. Zusätzliche Integritätsüberprüfungen (zum Beispiel auf Grundlage von nicht-interaktiven Zero-Knowledge-Beweisen) sind möglichen, erfordern aber zusätzlichen Rechenaufwand. Alternativ gibt es erweiterte homomorphe Verfahren, die eine eingeschränkte Menge von berechenbaren Funktionen erlauben [BSW12] oder von Benutzerseite ein Token zur effizienten Berechnung auf den verschlüsselten Daten benötigen [De17].

6 Zusammenfassung

Effiziente und sichere Datenhaltung in der Cloud ist ein zukunftsrelevantes Thema. Insbesondere Cloudatenbanken spielen dabei eine große Rolle, weil sie umfangreiche Funktionalitäten zur Datenverarbeitung anbieten. Zusammenfassend lässt sich sagen, dass eine externe Speicherung von stark verschlüsselten Daten durch einen Benutzer nicht sinnvoll ist, wenn er bei jedem Lesezugriff die Daten komplett herunterladen, entschlüsseln und durchsuchen muss. Daher müssen Datenbanken, die für Cloudspeicher-Anwendungen genutzt werden, entsprechend verschlüsselte Daten (vor-)verarbeiten können.
Auf Grundlage der hier gegebenen Übersicht und Anforderungsanalyse werden wir in zukünftigen Arbeiten das bestehende System um homomorphe Verfahren erweitern. Ein wesentlicher Arbeitsschritt wird es sein, sichere und effiziente Kombinationen von eigenschaftsbewahrenden und homomorphen Verschlüsselungsverfahren zu ermitteln.

Danksagung: Tim Waage und Daniel Homann wurden zum Teil finanziert durch die Deutsche Forschungsgemeinschaft im Projekt Wi 4086/2-2.

Literatur

[ABD16] Albrecht, M. R.; Bai, S.; Ducas, L.: A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes. In: CRYPTO 2016. Bd. 9814. LNCS, Springer, S. 153–178, 2016.

[Ar13] Arasu, A.; Blanas, S.; Eguro, K.; Joglekar, M.; Kaushik, R.; Kossmann, D.; Ramamurthy, R.; Upadhyaya, P.; Venkatesan, R.: Secure database-as-a-service with Cipherbase. In: SIGMOD 2013. ACM, S. 1033–1036, 2013.

[BGV14] Brakerski, Z.; Gentry, C.; Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption without Bootstrapping. TOCT 6/3, 13:1–13:36, 2014.

[Bo13] Bos, J. W.; Lauter, K. E.; Loftus, J.; Naehrig, M.: Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. In: Cryptography and Coding 2013. Bd. 8308. LNCS, Springer, S. 45–64, 2013.

[Br] Brenner, M.; Dai, W.; Halevi, S.; Han, K.; Jalali, A.; Kim, M.; Laine, K.; Malozemoff, A.; Paillier, P.; Polyakov, Y. et al.: A standard API for RLWE-based homomorphic encryption, http://homomorphicencryption.org/white_papers/API_homomorphic_encryption_white_paper.pdf.

[Br12] Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: CRYPTO 2012. Bd. 7417. LNCS, Springer, S. 868–886, 2012.

[BS13] Brenner, M.; Smith, M.: Caching Oblivious Memory Access: An Extension to the HCRYPT Virtual Machine. In: CCS 2013. ACM, S. 1363–1366, 2013.

[BS14] Bajaj, S.; Sion, R.: TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality. IEEE Trans. Knowl. Data Eng. 26/3, S. 752–765, 2014.

[BSW12] Boneh, D.; Segev, G.; Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ACM, S. 350–366, 2012.

[Ca15] Canteaut, A.; Carpov, S.; Fontaine, C.; Lepoint, T.; Naya-Plasencia, M.; Paillier, P.; Sirdey, R.: How to Compress Homomorphic Ciphertexts. IACR Cryptology ePrint Archive 2015/113, 2015.

[CJL16] Cheon, J. H.; Jeong, J.; Lee, C.: An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. LMS Journal of Computation and Mathematics 19/A, S. 255–266, 2016.

[CL15] Castagnos, G.; Laguillaumie, F.: Linearly Homomorphic Encryption from DDH. In: RSA Conference. Bd. 9048. LNCS, Springer, S. 487–505, 2015.

[CLT14] Coron, J.-S.; Lepoint, T.; Tibouchi, M.: Scale-Invariant Fully Homomorphic Encryption over the Integers. In: Public-Key Cryptography 2014. Bd. 8383. LNCS, Springer, S. 311–328, 2014.

[CS16] Costache, A.; Smart, N. P.: Which Ring Based Somewhat Homomorphic Encryption Scheme is Best? In: RSA Conference 2016. Bd. 9610. LNCS, Springer, S. 325–340, 2016.

[CS98] Cramer, R.; Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: CRYPTO 1998. Bd. 1462. LNCS, Springer, S. 13–25, 1998.

[De10] Desmedt, Y.; Gennaro, R.; Kurosawa, K.; Shoup, V.: A New and Improved Paradigm for Hybrid Encryption Secure Against Chosen-Ciphertext Attack. J. Cryptology 23/1, S. 91–120, 2010.

[De17] Desmedt, Y.; Iovino, V.; Persiano, G.; Visconti, I.: Controlled homomorphic encryption: definition and construction. In: International Conference on Financial Cryptography and Data Security. Springer, S. 107–129, 2017.

[Di10] van Dijk, M.; Gentry, C.; Halevi, S.; Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: EUROCRYPT 2010. Bd. 6110. LNCS, Springer, S. 24–43, 2010.

[FLA11] Fousse, L.; Lafourcade, P.; Alnuaimi, M.: Benaloh's Dense Probabilistic Encryption Revisited. In: AFRICACRYPT 2011. Bd. 6737. LNCS, Springer, S. 348–362, 2011.

[FV12] Fan, J.; Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption. IACR Cryptology ePrint Archive 2012/144, 2012.

[Ga84] Gamal, T. E.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: CRYPTO 1984. Bd. 196. LNCS, Springer, S. 10–18, 1984.

[Ge09] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009. ACM, S. 169–178, 2009.

[Ge10] Gentry, C.: Computing arbitrary functions of encrypted data. Commun. ACM 53/3, S. 97–105, 2010.

[GHS12] Gentry, C.; Halevi, S.; Smart, N. P.: Homomorphic Evaluation of the AES Circuit. In: CRYPTO 2012. Bd. 7417. LNCS, Springer, S. 850–867, 2012.

[GM84] Goldwasser, S.; Micali, S.: Probabilistic Encryption. J. Comput. Syst. Sci. 28/2, S. 270–299, 1984.

[HPS98] Hoffstein, J.; Pipher, J.; Silverman, J. H.: NTRU: A ring-based public key cryptosystem. In: International Algorithmic Number Theory Symposium. Springer, S. 267–288, 1998.

[JL13] Joye, M.; Libert, B.: Efficient Cryptosystems from 2^k-th Power Residue Symbols. In: EUROCRYPT 2013. Bd. 7881. LNCS, Springer, S. 76–92, 2013.

[Ke15] Kerschbaum, F.: Frequency-hiding order-preserving encryption. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, S. 656–667, 2015.

[KF16] Kirchner, P.; Fouque, P.-A.: Comparison between Subfield and Straightforward Attacks on NTRU. IACR Cryptology ePrint Archive 2016/717, 2016.

[KL15] Kim, M.; Lauter, K.: Private genome analysis through homomorphic encryption. BMC medical informatics and decision making 15/5, S3, 2015.

[Ko87] Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation 48/177, S. 203–209, 1987.

[LN14] Lepoint, T.; Naehrig, M.: A Comparison of the Homomorphic Encryption Schemes FV and YASHE. In: AFRICACRYPT 2014. Bd. 8469. LNCS, Springer, S. 318–335, 2014.

[LTV17] López-Alt, A.; Tromer, E.; Vaikuntanathan, V.: Multikey Fully Homomorphic Encryption and Applications. SIAM Journal on Computing 46/6, S. 1827–1892, 2017.

[NKW15] Naveed, M.; Kamara, S.; Wright, C. V.: Inference attacks on property-preserving encrypted databases. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, S. 644–655, 2015.

[NLV11] Naehrig, M.; Lauter, K. E.; Vaikuntanathan, V.: Can homomorphic encryption be practical? In: Cloud Computing Security Workshop (CCSW) 2011. ACM, S. 113–124, 2011.

[Pa99] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: EUROCRYPT 1999. Bd. 1592. LNCS, Springer, S. 223–238, 1999.

[PBS11] Perl, H.; Brenner, M.; Smith, M.: Poster: An Implementation of the Fully Homomorphic Smart-Vercauteren Cryptosystem. In: CCS 2011. ACM, S. 837–840, 2011.

[Po12] Popa, R. A.; Redfield, C. M. S.; Zeldovich, N.; Balakrishnan, H.: CryptDB: processing queries on an encrypted database. Commun. ACM 55/9, S. 103–111, 2012.

[PZB15] Popa, R. A.; Zeldovich, N.; Balakrishnan, H.: Guidelines for Using the CryptDB System Securely. IACR Cryptology ePrint Archive 2015/979, 2015.

[RAD78] Rivest, R. L.; Adleman, L.; Dertouzos, M. L.: On data banks and privacy homomorphisms. Foundations of secure computation 4/11, S. 169–180, 1978.

[Sh17] Shameer, K.; Badgeley, M. A.; Miotto, R.; Glicksberg, B. S.; Morgan, J. W.; Dudley, J. T.: Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams. Briefings in bioinformatics 18/1, S. 105–124, 2017.

[Sm14] Smart, N. P.; Rijmen, V.; Gierlichs, B.; Paterson, K.; Stam, M.; Warinschi, B.; Watson, G.: Algorithms, key size and parameters report, Techn. Ber., European Union Agency for Network und Information Security, 2014.

[Tu13] Tu, S.; Kaashoek, M. F.; Madden, S.; Zeldovich, N.: Processing Analytical Queries over Encrypted Data. VLDB 6/5, S. 289–300, 2013.

[Wi15] Wiese, L.: Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases. Walter de Gruyter GmbH & Co KG, 2015.

[WW17] Waage, T.; Wiese, L.: Property Preserving Encryption in NoSQL Wide Column Stores. In: Cloud and Trusted Computing 2017. Bd. 10574. LNCS, Springer, S. 3–21, 2017.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 235

Informationssicherheitskonzept nach IT-Grundschutz für Containervirtualisierung in der Cloud

Erik Buchmann[1], Andreas Hartmann[2], Stephanie Bauer[3]

Abstract:

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) stellt mit dem IT-Grundschutz eine sichere und wirksame Schutzvorkehrung vor den stetig steigenden Bedrohungen im Kontext der Digitalisierung zur Verfügung. Zwar sind die behandelten BSI-Bausteine herstellerneutral definiert. Gleichwohl beziehen sich die Bausteine auf die sich ändernden Technologien, was eine entsprechende Anpassung erforderlich macht. Mit dem Hintergrund von Cloud basierten IT-Infrastrukturen findet aktuell ein massiver Wandel hinsichtlich eingesetzter Servertechnologien und –dienste hin zu Containervirtualisierung in der Cloud statt. Unternehmen, die ihre IT-Landschaften diesbezüglich transformieren, müssen darum mehr denn je die Sicherheit ihrer Daten gewährleisten. Wir zeigen am Beispiel von Docker Containern, wie der IT-Grundschutz auf diese neuen Herausforderungen anzupassen ist. Wir gehen dabei insbesondere auf die Gefährdungsanalyse, Docker-spezifische Gefährdungen sowie entsprechende Maßnahmen ein.

Keywords: IT-Grundschutz; Digitalisierung; IT-Sicherheit; Docker Container; Cloud Technologien

1 Einleitung

Die Containervirtualisierung, z.B. mit dem Open-Source Projekt Docker [Do17], ermöglicht es durch die Bereitstellung von Containern, innovative und cloudbasierte Anwendungen auf eine agile, kosteneffiziente Weise umzusetzen und bereitzustellen. Um dabei Tempo- und Innovationsvorteile zu realisieren müssen Unternehmen ihre traditionelle IT-Infrastruktur anpassen [Gö17]. Dies ist zwingend mit einer Anpassung des IT-Sicherheitskonzepts verbunden. Wenn das Sicherheitskonzept eines Unternehmens auf dem IT-Grundschutz [Bu11] des Bundesamts für Sicherheit in der Informationstechnik (BSI) beruht oder im Rahmen einer ISO 27001 Zertifizierung auf dem IT-Grundschutz aufgebaut [Bu14] wurde, ist dies schwierig: Ein Baustein für die Containervirtualisierung mittels Docker oder alternativer Produkte existiert derzeit nicht. Allerdings gibt es Grundlagen, z.B. einen Baustein B 3.304 für den sicheren Betrieb von Virtualisierungsservern oder einen Katalog von allgemeingültigen Elementargefährdungen.

In dieser Arbeit untersuchen wir, inwiefern der bestehende IT-Grundschutz nach BSI auf die Containervirtualsierung mit Docker angewendet werden kann, um die Virtualisierungsinfrastruktur vom physischen Server bis zu den Containern abzusichern. Unser wesentlicher

[1] Hochschule für Telekommunikation Leipzig, Deutschland, buchmann@hft-leipzig.de
[2] Hochschule für Telekommunikation Leipzig, Deutschland, hartmann@hft-leipzig.de
[3] T-Systems International GmbH, Nürnberg, Deutschland, stephanie.bauer@telekom.de

doi:10.18420/sicherheit2018_18

Beitrag ist eine Gefährdungsanalyse für alle Komponenten des Docker-Ökosystems, die mit den Bausteinen des BSI nicht ausreichend abgesichert werden. Das heißt, wir beschreiben Docker-spezifische Gefährdungen und entsprechende Maßnahmen zum Umgang mit diesen Gefährdungen, die über die BSI Grundschutz-Kataloge hinausgehen. Zuletzt diskutieren wir, ob diese Aspekten in einen benutzerdefinierten Baustein [Bu17a] für die Containervirtualisierung integriert werden sollten.

Unsere Untersuchung erfolgt für eine On-Premise-Lösung, bei der der Nutzer Eigentümer der Infrastruktur und der Container-Plattform ist. Übergreifende Aspekte, die sich auf die komplette Organisation auswirken, sowie sicherheitstechnische Aspekte bzgl. der Infrastruktur und der Netze werden zentral gesteuert. Wir gehen davon aus, dass dafür bereits ein Sicherheitskonzept nach IT-Grundschutz vorliegt. In unserem Fokus liegt daher die Schicht 3 *IT-Systeme* des Grundschutzes, auf der das Docker-Ökosystem angesiedelt ist. Unsere vollständige Untersuchung kann unter [Ba17] abgerufen werden.

Aufbau der Arbeit: Abschnitt 2 beschreibt die Grundlagen dieser Arbeit. In Abschnitt 3 führen wir eine Risikoanalyse für Docker nach IT-Grundschutz durch und beschreiben spezifische Gefährdungen und Maßnahmen. Die Arbeit schließt mit einem Fazit in Abschnitt 4.

2 Grundlagen

In diesem Abschnitt stellen wir Docker Container [PR15] sowie die Vorgehensweise des IT-Grundschutzes nach BSI kurz vor.

2.1 Docker Container

Die Containervirtualisierung auf der Betriebssystemebene eines Linux-Systems kapselt Applikationen in Containern. Dies ermöglicht es, Systemressourcen wie Prozessor, Netzwerk oder Speicher zu verwalten, Applikationen über Systeme hinweg zu verschieben oder voneinander isolierte Container parallel auf dem selben Host-System zu betreiben. Dabei setzen die Container auf Funktionen des Linux-Kernels auf. Abbildung 1 zeigt einen typischen Docker-Aufbau.

Die Docker Architektur [Do17] besteht aus Docker Client, Docker Daemon, Docker Registry und den Docker Objekten (Images, Docker Files, Container). Der **Docker Client** nimmt Anweisungen des Anwenders entgegen. Der **Docker Daemon** stellt Systemfunktionen zum Erstellen, Betreiben und Verteilen von Containern zur Verfügung. Diese beiden Bestandteile bilden zusammen die Docker Engine. Die Docker-Engine nutzen drei voneinander isolierte Container. Ein Container besteht aus zwei Hauptverzeichnissen: `/bin` enthält die Binärdateien und `/lib` die dynamischen Bibliotheken und Kernel-Module, die für die Funktionalität eines Containers benötigt werden.

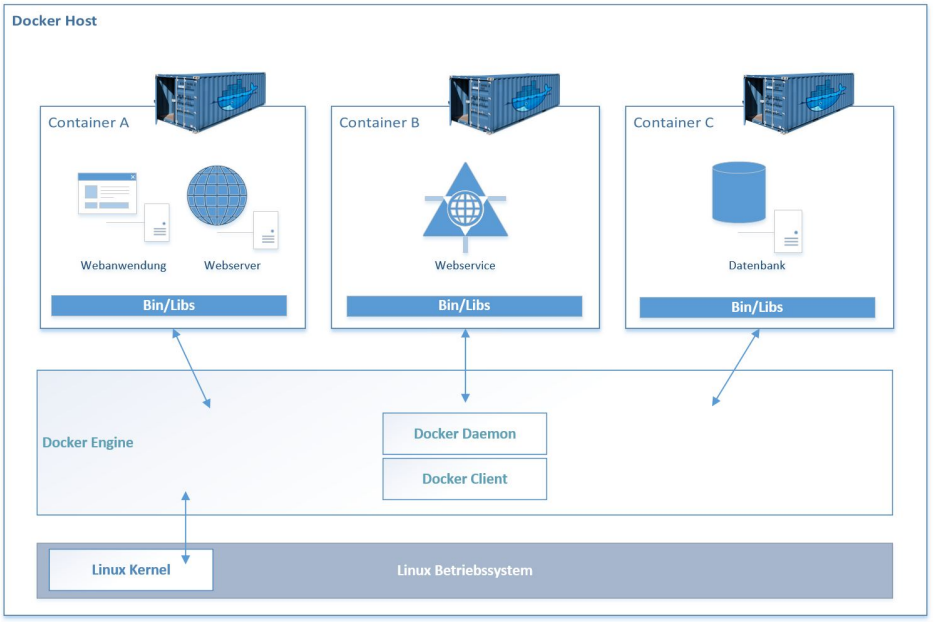


Abb. 1: Docker-Verbund

Client und Daemon können auf dem gleichen Host-System laufen oder der Client wird mit einem remote Daemon verbunden. Die Kommunikation findet über eine REST API, ein UNIX Socket oder eine andere Netzwerkschnittstelle statt. Wenn die Installation auf unterschiedlichen Systemen erfolgt, muss die Konfiguration bzgl. der jeweiligen Netzwerkinformationen manuell vorgenommen werden. Die **Docker Registry** verwaltet und speichert die Images. Docker Hub und Docker Cloud sind bspw. öffentlich zugängliche Registries. Docker ist in seinen Grundeinstellungen so konfiguriert, dass nach Images aus dem Docker Hub gesucht wird. Es besteht aber auch die Möglichkeit, eine private Registry für seine Images anzulegen, z.B. über die Docker Trusted Registry.

Docker **Images** sind schreibgeschützte Templates für die Erstellung eines Containers, d.h, ein **Container** ist die ausführbare Instanz eines Images. Ein Docker-Image besteht aus Layern: Einem Base Image mit einem Debian-Betriebssystem sowie mehreren Layern, die z.B. einem Apache Web-Server oder Anwendungen enthalten. Die Layer werden dabei mittels Union Mount über das Base Image „darübergelegt“. Das Advanced Unification Filesystem stellt dafür eine Copy-On-Write Funktion zur Verfügung. Diese Unterscheidung ist für das Sicherheitskonzept wichtig, da das Base Image oft aus öffentlichen Quellen stammt, während die übergeordneten Layer anwendungsspezifisch erstellt werden.

2.2 IT-Grundschutz nach BSI

Das BSI hat mit dem IT-Grundschutz ein erprobtes Verfahren zur Herstellung eines ausgewogenen Sicherheitsniveaus entwickelt, das aus vier Standards besteht und den Weg in eine ISO 27001-Zertifizierung ebnet:

- BSI-Standard 200-1: Management-Systeme für Informationssicherheit
- BSI-Standard 200-2: IT-Grundschutz-Vorgehensweise
- BSI-Standard 200-3: Risikomanagement
- BSI Standard 100-4: Notfallmanagement

Vor der Modernisierung des IT-Grundschutzes im November 2017 verwiesen die BSI-Standards auf die IT-Grundschutz-Kataloge [Bu16], welche Standard-Sicherheitsmaßnahmen für typische Geschäftsprozesse, Anwendungen und IT-Systeme enthalten. In den aktuellen Standards wird stattdessen auf das IT-Grundschutz-Kompendium [Bu17b] verwiesen, das zwischen Basis-, Standard- und erhöhten Anforderungen an die IT-Sicherheit differenziert und ein flexibleres Risikomanagement inklusive nutzerdefinierter Schutzprofile und Bausteine unterstützt.

Unsere Arbeit basiert noch auf den BSI-Standards 100-1 bis 100-3 und der Ergänzung zum BSI-Standard 100-3 sowie den IT-Grundschutzkatalogen Stand 15. Ergänzungslieferung [Bu08a, Bu08b, Bu08c, Bu16]. Eine Übertragung unserer Erkenntnisse auf die Aktualisierung des IT-Grundschutzes ist jedoch gegeben: Zwar haben sich durch die neue Untergliederung des Bausteinkatalogs die Namen der von uns verwendeten Bausteine geändert. Sie sind jedoch hinreichend unverändert Bestandteil des IT-Grundschutz-Kompendiums (vgl. Baustein „B 3.304 Virtualisierung“ und „SYS.1.5 Virtualisierung“). Darüber hinaus haben wir keine Restrisiken als „akzeptabel“ bewertet – der neue Standard 200-3 legt sieht die Akzeptanz von Risiken eine andere Vorgehensweise vor als 100-3.

3 IT-Grundschutz für Docker-Container

Der IT-Grundschutz erfordert eine Definition des Informationsverbundes, damit der Geltungsbereich des Sicherheitskonzepts festgelegt werden kann. Wir beschränken uns hier auf das Docker-Ökosystem innerhalb der Schicht *IT-Systeme*. Wir legen dafür einen typischen Docker-Aufbau zugrunde, wie er in Abbildung 1 dargestellt ist. In Container A läuft eine Webanwendung auf einem Webserver. In Container B läuft ein Webservice, beispielsweise die Zahlungsmethode in einem E-Shop. In Container C wird eine Datenbank betrieben, die wiederum Logindaten beinhaltet. Diese drei Container werden isoliert voneinander betrieben, aber müssen miteinander kommunizieren. Wir haben den Informationsverbund in Anhang A zusammengefasst. Details finden sich in [Ba17].

Aus dem Informationsverbund lässt sich ableiten, dass die existierenden BSI-Bausteine die IT-Anwendungen und die eingesetzte Server-Software bereits ausreichend absichern (s. Anhang B und C). Unsere zentrale Erkenntnis ist, dass für die Server-Systeme, auf denen Docker

abläuft, ein hoher Schutzbedarf besteht. Wir führen daher im Folgenden eine Risikoanalyse nach IT-Grundschutz für diese Systeme durch. Am Ende dieses Abschnitts diskutieren wir, inwiefern sich unsere Erkenntnisse von Docker auf die Containervirtualisierung im Allgemeinen und auf andere Anwendungsszenarien übertragen lassen.

3.1 Risikoanalyse für Server-Systeme mit Docker-Containern

Für die Docker Host-Komponente sind 9 Elementargefährdungen relevant:

- G 0.15 Abhören
- G 0.18 Fehlplanung oder fehlende Anpassung
- G 0.19 Offenlegung von Informationen
- G 0.21 Manipulation von Hard-oder Software
- G 0.23 Unbefugtes Eindringen
- G 0.27 Ressourcenmangel
- G 0.32 Rechtemissbrauch
- G 0.40 Verhinderung von Diensten
- G 0.46 Integritätsverlust

Nachfolgend wird für jede Gefährdung eine Maßnahme aus dem Maßnahmen-Katalog festgelegt und bewertet. Wenn alle Bewertungskriterien mit „J“ gekennzeichnet werden, ist die Maßnahme ausreichend und eine Risikobehandlung kann entfallen. Anderenfalls wird die spezielle Gefährdungslage genauer untersucht und ergänzende Sicherheitsmaßnahmen vorgeschlagen. Ausführliche Begründungen für die Bewertungen können in [Ba17] eingesehen werden.

3.1.1 G 0.15 Abhören

Maßnahme	M 5.177 Serverseitige Verwendung von SSL/TLS	
Bewertung	Vollständigkeit vorhanden: J Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: J
Begründung der Bewertung	Da die Datenobjekte D1 Personendaten, D2 Nutzdaten und D3 Accountdaten einen hohen Schutzbedarf bzgl. Vertraulichkeit aufweisen, muss sichergestellt werden, dass diese Datenobjekte über die Kommunikationsverbindung Kom2 nicht abgehört werden können. Dies wird durch serverseitiges SSL/TLS erreicht.	

3.1.2 G 0.18 Fehlplanung oder fehlende Anpassung

Maßnahme	M 2.38 Aufteilung der Administrationstätigkeiten M 2.446 Aufteilung der Administration bei Virtualisierungsserven

Bewertung	Vollständigkeit vorhanden: N Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: N
Begründung der Bewertung	Die Administration virtueller IT-Systeme geht über die in M 2.38 und M 2.446 beschriebenen Rollen hinaus.	

Spezifische Gefährdungen / Maßnahmen: Ein Gefährdungsszenario ist der *Container Breakout*, der einem Angreifer Zugriff auf das Host-System oder auf weitere Container im gleichen System erlaubt, und zwar mit den Privilegien[4] des Containers, aus dem der Ausbruch erfolgte.

- **Definition der Systembenutzer** Docker-Container sind grundsätzlich nicht als privilegierte Container zu betreiben, damit Angreifer im Erfolgsfall nur unprivilegierten Zugriff auf andere Ressourcen erhalten.
- **Rechtemanagement** Es sind die Berechtigungen für die definierten Benutzergruppen auf Minimalität zu prüfen.
- **Rollenaufteilung** Wenn in der Maßnahme M 2.38 eine Rollenaufteilung der Administratoren festgelegt wird, ist zu prüfen, ob auch für virtuelle IT-Systeme eine Aufteilung notwendig ist.
- **Weitere Linux Funktionen** Schutzmaßnahmen wie apparmor, selinux, seccomp, filter und namespaces, die auf dem Host-System installiert werden, können das Risiko eines Ausbruchs aus einem Gastcontainer reduzieren.

3.1.3 G 0.19 Offenlegung schützenswerter Informationen

Maßnahme	M 5.177 Serverseitige Verwendung von SSL/TLS	
Bewertung	Vollständigkeit vorhanden: J Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: J
Begründung der Bewertung	Für hoch vertrauliche Daten (D1 Personendaten, D2 Nutzdaten, D3 Accountdaten), besteht die Gefahr einer Offenlegung durch technisches Versagen oder vorsätzliche Handlungen. Eine Absicherung durch SSL/TLS reduziert dieses Sicherheitsrisiko.	

Hinweis: Da unsere Risikoanalyse auf Docker-Container ausgerichtet ist, betrifft die Einschätzung der Maßnahme „Serverseitige Verwendung von SSL/TLS“ nur die Kommunikation der Docker-Container untereinander. Für eine Kommunikation nach außen ist unter Umständen eine separate Risikoanalsye erforderlich, ebenso für externe Aspekte wie beispielsweise ein ausgelagertes Speicher-Subsystem.

[4] Das Docker-Team arbeitet daran, den root-Benutzer in einem Container automatisch auf einen Nicht-Root-Benutzer im Host-System abzubilden (Reduzierung der Auswirkungen eines Ausbruchs).

3.1.4 G 0.21 Manipulation von Hard-oder Software

Maßnahme	M 2.448 Überwachung virtueller Infrastrukturen	
Bewertung	Vollständigkeit vorhanden: N Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: N
Begründung der Bewertung	Durch zentrale Speicherung von Images im dem Docker Hub ergeben sich Angriffspunkte für die Manipulation von Software, die es bei bisheriger Virtualisierungstechnologie nicht gab.	

Spezifische Gefährdungen / Maßnahmen: Eine besondere Gefährdung durch Manipulation für Docker bei Container ist der Einsatz von *vergifteten Images*, d.h., Images aus öffentlicher Quelle, die einen Schadcode enthalten.

- **Docker Content Trust** Dies ist ein Feature von Docker, durch das Signieren mit dem öffentlichen Schlüssel des Entwicklers die Integrität der Images zu sichern und Image-Erzeuger zu schützen.
- **Docker Trusted Registry** Die Docker Trusted Registry bietet die Möglichkeit, die Images unter Verzicht einer öffentlich zugänglichen Registry on-premises oder in einer virtuellen private Cloud zu speichern und zu verwalten.

3.1.5 G 0.23 Unbefugtes Eindringen in IT-Systeme

Maßnahme	M 2.448 Überwachung virtueller Infrastrukturen M 4.346 Sichere Konfiguration virtueller IT-Systeme M 5.154 Sichere Konfiguration eines Netzes für virtuelle Infrastruktur	
Bewertung	Vollständigkeit vorhanden: N Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: N
Begründung der Bewertung	Veränderungen an den Binärdateien wirken sich bei der Betriebssystemvirtualisierung auf alle Container aus und nicht nur auf den Virtualisierungsserver selbst.	

Spezifische Gefährdungen / Maßnahmen: Durch *unautorisierte Änderungen an Konfigurationsdateien* der virtuellen Infrastruktur können erhebliche und tiefgreifende Schäden entstehen, ebenso wie durch vorsätzliche oder versehentliche Fehlkonfigurationen der Netzzuordnung. Hier stellt insbesondere der *Docker Daemon* eine Angriffsoberfläche dar, da dieser root-Rechte besitzt und die Funktionsfähigkeit aller Container beeinflussen kann. Für Vertraulichkeit, Integrität oder Verfügbarkeit der Daten ist die *Integrität von Konfigurationsdaten* ausschlaggebend.

Für den Schutz des Docker Daemons sind mehrere aufeinander abgestimmte Maßnahmen erforderlich, die das Rollen- und Rechtemanagement betreffen, die Manipulation von Hard- oder Software sowie den Missbrauch von Berechtigungen berücksichtigen, sowie die Konfiguration des Host-Systems und des virtuellen Netzwerks absichern.

- **Prüfsummen** Die Prüfung auf unautorisierte Änderungen der Konfigurationsdateien kann beispielsweise mittels Werkzeugen wie OS-SEC erfolgen.
- **Docker Bench for Security** Docker selbst bietet das Docker Bench for Security Script [Ce17] an, welches die eigene Docker Konfiguration prüft. Voraussetzung ist eine Dockerversion 1.10.0 oder aktueller.
- **Konfiguration der Netzfunktionen** Da Docker Container auf einem gängigen Linux-System betrieben werden, kann man auf bekannte Werkzeuge wie beispielsweise Puppet [Je17] zurückgreifen, um die Netzkomponenten zentral zu überwachen.
- **Benennung virtueller Netze** Wenn Netzverbindungen auf verschiedenen Host-Systemen gleich benannt sind, kann ein Container versehentlich mit dem falschen Netzwerk verbunden werden. Eine eindeutige und aussagekräftige Benennung der Netze sollte anhand der Funktion des Netzwerkes vorgenommen werden [Je17].
- **Storage Zentralisierung** Im Sicherheitskonzept muss festgelegt werden, ob Daten nach Beenden des Containers gelöscht werden oder ob ein Dateiverzeichnis des Containers auf ein Dateiverzeichnis des Host-Systems verknüpft wird. Das Host-System muss dann für eine Abgrenzung zwischen den Daten des Betriebssystemkerns, der Systembibliotheken und der gemeinsam genutzten Anwendungen sichergestellen.
- **Monitoring** Das Monitoring lässt sich durch den Einsatz eines Linux-Servers mit den systemeigenen Monitoring-Systemen wie Nagios bewerkstelligen [Je17].
- **Kommunikation zwischen Containern** Wird das Container Linking aktiviert, so müssen Container, die nicht miteinander kommunizieren dürfen, durch Firewalls oder physikalische Trennung voneinander isoliert werden. Maßnahme M 5.154 Sichere Konfiguration eines Netzes für virtuelle Infrastruktur bietet hierzu eine Grundlage.

3.1.6 G 0.27 Ressourcenmangel

Maßnahme	M 4.349 Sicherer Betrieb von virtuellen Infrastrukturen	
Bewertung	Vollständigkeit vorhanden: J Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: J
Begründung der Bewertung	Bei dem Betrieb von einem Container Host-System sind die gleichen Dinge zu beachten wie bei einem Linux-Server. Dadurch greifen die üblichen Schutzmaßnahmen für den Zugriff auf einen Linux-Server.	

3.1.7 G 0.32 Missbrauch von Berechtigungen

Maßnahme	M 2.318 Sichere Installation eines IT-Systems M 2.444 Einsatzplanung für virtuelle IT-Systeme M 2.447 Sicherer Einsatz virtueller IT-Systeme M 3.72 Grundbegriffe der Virtualisierungstechnik	
Bewertung	Vollständigkeit vorhanden: J Mechanismenstärke ausreichend: N	Zuverlässigkeit gegeben: J Ausreichender Schutz: N
Begründung der Bewertung	Eine bedeutende Gefährdung für Container ist der Missbrauch von Rechten, insbesondere wenn Angreifer root-Rechte erlangen.	

Spezifische Gefährdungen / Maßnahmen: Da Container auf Kernel-Funktionen des Host-Systems zugreifen, können Angreifer mit *unrechtmäßig erworbenen* Berechtigungen großen Schaden anrichten. Gleiches gilt für den *Mißbrauch* von rechtmäßige Berechtigungen.

Neben den bereits unter G 0.18 Fehlplanung oder fehlende Anpassung aufgeführten Maßnahmen sollten folgende Hinweise beachtet werden:

- **Isolierung und Kapselung** Ein Schutz gegen den Missbrauch von Berechtigungen wird durch Isolierung und Kapselung von virtuellen IT-Systemen realisiert, beispielsweise mittels namespaces und cgroups.
- **Schulung der Administratoren** Da die Container-Technologie einer dynamischen Entwicklung unterliegt, kommt einer regelmäßigen Schulung große Bedeutung zu.

3.1.8 G 0.40 Verhinderung von Diensten (DoS)

Maßnahme	M 4.405 Verhinderung von DoS bei Webanwend. und -Services M 4.97 Ein Dienst pro Server	
Bewertung	Vollständigkeit vorhanden: N Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: N
Begründung der Bewertung	Da sich mehrere Container die Ressourcen eines Systems teilen können, erhöht sich das Risiko durch einen Denial-of-Service-Angriff.	

Spezifische Gefährdungen / Maßnahmen: Bei der Containervirtualisierung kann es neben klassischen *Denial-of-Service-Angriffen* auch zu einer *Überbuchung von Ressourcen* kommen, wenn die einzelnen Container durch Manipulation oder Fehlkonfiguration in Summe mehr Ressourcen zugewiesen bekommen, als physisch auf dem Host-System vorhanden sind.

- **Capabilities** Um eine Überbuchung zu verhindern, kann durch Linux-capabilities (limits, cgroups) der Zugriff der Containern auf CPU, Arbeitsspeicher, etc. limitiert werden.
- **Ein Dienst pro Server** Um die Auswirkungen eines Angriffs zu minimieren, sollte Maßnahme M 4.97 *Ein Dienst pro Server* in Betracht gezogen werden.

3.1.9 G 0.46 Integritätsverlust schützenswerter Information

Maßnahme	M 5.177 Serverseitige Verwendung von SSL/TLS	
Bewertung	Vollständigkeit vorhanden: J Mechanismenstärke ausreichend: J	Zuverlässigkeit gegeben: J Ausreichender Schutz: J
Begründung der Bewertung	Durch Manipulationen, Fehlverhalten, Fehlfunktionen etc. kann die Datenintegrität beeinträchtigt werden. SSL/TLS unterbindet dies.	

3.2 Diskussion

Erkenntnisse aus der Risikoanalyse: Der Aufbau des *IT-Verbunds* ergibt sich wesentlich aus dem Docker-Ökosystem, d.h., hier bestehen keine wesentlichen Änderungsmöglichkeiten. Beispielsweise ändern sich unsere Aussagen nicht, wenn der IT-Verbund statt in zwei in drei Rechenzentren parallel betrieben wird.

Mit den Datenobjekten D1 Personendaten, D2 Nutzdaten, D3 Accountdaten und D4 Konfigurationsdaten hat der IT-Verbund jeweils mindestens ein Datenobjekt, das einen hohen Schutzbedarf bzgl. Vertraulichkeit, Integrität und Verfügbarkeit aufweist. Da sich stets der höchste Schutzbedarf der Datenobjekte auf die Anwendungen und Systeme im IT-Verbund vererbt, hat die konkrete Wahl der Anwendung keinen Einfluss auf das Sicherheitskonzept, von Extremfällen (z.B. nicht-geschäftskritische Testplattformen ohne Außenverbindung oder das interne Logging von Prozessdaten ohne Personenbezug und Vertraulichkeitsanforderungen) abgesehen.

IT-Grundschutz für Containervirtualisierung: Neben den Maßnahmen, die sich aus dem Schutzbedarf der genannten Datenobjekte ableiten lassen, haben wir Maßnahmen identifiziert, die sich auf das Docker-Ökosystem beziehen. Diese Maßnahmen sind nicht Bestandteil der existierenden BSI-Bausteine. Sie sind jedoch für ein Sicherheitskonzept für die Containervirtualisierung unverzichtbar.

Zwar lassen sich aus dem Virtualisierungsbaustein B 3.304 einige Maßnahmen ableiten. Das BSI macht aber in der überarbeiteten Fassung SYS.1.5 dieses Bausteins im Grundschutz-Kompendium deutlich, dass dieser Baustein nicht für Docker-Container zu verwenden ist. Daher stellt sich die Frage, ob dafür ein benutzerdefinierter Baustein (s. [Bu17a]) durch die Nutzer der Containervirtualisierung erstellt werden sollte.

Die Containervirtualisierung ist eine junge Technologie, die noch stetig weiterentwickelt wird. Ein benutzerdefinierter Baustein hätte den Vorteil, dass dieser sehr rasch und unmittelbar durch die Domänenexperten erstellt werden könnte. Auf der anderen Seite wird die Containervirtualisierung in sehr heterogenen Umgebungen eingesetzt. Daher steht zu befürchten, dass Aufbau, Abstimmung und Pflege eines benutzerdefinierten Bausteins zu einem erheblichen und repetitivem Koordinationsaufwand unter den Anwendern führen würde, sodass Erweiterungen oder Korrekturen nur mit großem Zeitverzug umgesetzt werden könnten. Aus unserer Sicht ist daher die zentrale Aufnahme und Pflege eines generischen Bausteins „Containervirtualisierung“ durch das BSI die bessere Alternative.

4 Zusammenfassung

Das Ziel dieser Arbeit bestand in der Entwicklung eines IT-Sicherheitskonzepts nach BSI IT-Grundschutz für Docker Container auf einem physikalischen Host-System in einer On-Premise Umgebung, sowie in der Verallgemeinerung dieser Erkenntnisse. Wir können feststellen, dass die BSI Grundschutz-Kataloge eine wertvolle Hilfestellung bei der Erstellung eines Sicherheitskonzepts für Docker bieten, insbesondere wenn es sich um Maßnahmen handelt, die auf Grund des Schutzbedarfs der Datenobjekte umzusetzen sind.

Maßnahmen, die sich auf das Docker Ökosystem beziehen, sind jedoch entweder selbst zu erarbeiten oder aus Virtualisierungsbaustein B 3.304 abzuleiten, der dafür jedoch nicht gedacht ist. Daher würde sich die Erstellung eines generischen Virtualisierungsbausteins durch das BSI anbieten. In der Zwischenzeit ließe sich ein Sicherheitskonzept für die Containervirtualisierung auch mittels eines benutzerdefinierten Bausteins umsetzen.

Literaturverzeichnis

[Ba17] Bauer, Stephanie: Erarbeitung eines Informationssicherheitskonzepts nach IT-Grundschutz für Docker Container. Bachelor-Arbeit, Hochschule für Telekommunikation Leipzig, Kopie s. http://www.webcitation.org/6xAkE4g1l, 2017.

[Bu08a] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 100-1, Managementsysteme für Informationssicherheit (ISMS). https://www.bsi.bund.de, 2008.

[Bu08b] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 100-2, IT-Grundschutz-Vorgehnsweise. https://www.bsi.bund.de, 2008.

[Bu08c] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 100-3, Risikoanalyse auf der Basis von IT-Grundschutz. https://www.bsi.bund.de, 2008.

[Bu11] Bundesamt für Sicherheit in der Informationstechnik: Webkurs IT-Grundschutz, IT -Grundschutz im Selbststudium. https://www.bsi.bund.de, 2011.

[Bu14] Bundesamt für Sicherheit in der Informationstechnik: Zertifizierung nach ISO 27001 auf der Basis von IT-Grundschutz. https://www.bsi.bund.de, 2014.

[Bu16] Bundesamt für Sicherheit in der Informationstechnik: IT-Grundschutz-Kataloge, 15.Ergänzungslieferung - 2016. https://www.bsi.bund.de, 2016.

[Bu17a] Bundesamt für Sicherheit in der Informationstechnik: Autorenrichtlinie zur Erstellung eines benutzerdefinierten Bausteins. https://www.bsi.bund.de, 2017.

[Bu17b] Bundesamt für Sicherheit in der Informationstechnik: IT-Grundschutz-Kompendium 2018, 1. Edition. https://www.bsi.bund.de, 2017.

[Ce17] Center for Internet Security: Docker Community Edition Benchmark. https://www.cisecurity.org, 2017.

[Do17] Docker Inc.: Docker Overview. https://docs.docker.com/engine/docker-overview/, Kopie s. http://www.webcitation.org/6xE2rYUYa, 2017.

[Gö17] Göbel, Lars: Container-as-a-Service - Die Zukunft der Virtualisierung. Cloud Computing Insider, Kopie s. http://www.webcitation.org/6xE2RH90v, 2017.

[Je17] Jedecke, Daniel: IT-Grundschutz in LXC-Container verpackt - Gut separiert. ix - Magazin für professionelle Informationstechnik, 5:116–, 2017.

[PR15] Pethuru Raj, Jeeva S. Chelladhurai, Vinod Singh: Learning Docker. Packt Publishing, 2015.

A Informationsverbund und Strukturanalyse

Für die Schutzbedarfsfeststellung ist eine Erfassung der IT-Systeme, IT-Anwendungen und die Kommunikationsverbindungen erforderlich. Als kleinste Einheit werden die Daten im IT-Verbund erfasst (s. Tabelle 1).

Nr.	Datenobjekt	Anmerkung
D1	Personendaten	personenbezogene Daten gemäß § 3 Abs. 1 BDSG
D2	Nutzdaten	generische Fachdaten
D3	Accountdaten	Authentifizierung und Autorisierung
D4	Konfigurationsdaten	Konfigurationsdaten und Parameter inkl. DNS und NTP
D5	Protokolldaten	Technische Protokolldaten (Monitoring)

Tab. 1: Datenerfassung

In Tabelle 2 werden diese Daten den IT-Anwendungen im Informationsverbund zugeordnet. Mit Fokus auf Docker verzichten wir auf die eigenständige Erfassung des Betriebssystems.

Nr.	Beschreibung	verarbeitete Daten	Software
A1	Webanwendung	D1, D2, D3, D4, D5	Allgemeine Webanwendung z.B. PHP
A2	Webserver	D1, D2, D4, D5	Apache Webserver
A3	Webservice	D2, D3, D4, D5	REST-basierter Dienst
A4	Datenbank	D1, D2, D3, D4, D5	Allgemeine Datenbank z.B. MySQL

Tab. 2: Strukturanalyse IT-Anwendungen, Server-Software und Kommunikationsverbindung

Weiterhin bezeichnet „Server-Software“ die Software, die auf dem Host-System aufsetzt, und „Docker Software“ die Bestandteile der Docker Engine. S1 und S2 stehen für je einen RZ-Standort basierend auf einem x86 Linux-Server (aktiv) - hier ordnen wir D1, D2, D3, D4 und D5 zu. Die Docker-Software SSW1 verarbeitet Daten entsprechend D4 und D5 mit Bezug zu S1 und S2. Die Kritikalität der Kommunikationsverbindungen KOM1 (extern, D1, D2, D3, Internet:https) und KOM2 (intern, D1, D2, D3, D4, D5, S1-S2 bidirektional SSH) richtet sich nach (1) der Existenz einer Außenverbindung und (2) dem Schutzbedarf der zu übertragenden Datenobjekte. Die Schutzbedarfsfeststellung erfolgt gemäß BSI-Standard 100-2 anhand der verwendeten Datenobjekte, indem typische Schadensszenarien (Verstoß gegen Gesetze, Gesundheitsschäden, Beeinträchtigung der Aufgabenerfüllung, finanzielle Auswirkungen, etc.) zugrundegelegt werden.

B Schutzbedarfsfeststellung

Die Schutzbedarfsfeststellung erfolgt gemäß BSI-Standard 100-2 anhand der Datenobjekte (Tabelle 3), indem typische Schadensszenarien (Verstoß gegen Gesetze, Gesundheitsschäden, Beeinträchtigung der Aufgabenerfüllung, etc.) zugrundegelegt werden.

Komponente	Vertraulichkeit	Integrität	Verfügbarkeit
D1: Personendaten	hoch	normal	normal
D2: Nutzdaten	hoch	normal	hoch
D3: Accountdaten	hoch	normal	normal
D4: Konfigurationsdaten	normal	hoch	normal
D5: Protokolldaten	normal	normal	normal

Tab. 3: Ergebnisse der Schutzbedarfsfeststellung für Datenobjekte

Der Schutzbedarf für eine IT-Anwendung entspricht dem höchsten Schutzbedarf der von der Anwendung verarbeiteten Datenobjekte. Äquivalent werden die Schutzbedarfe für die Server-Software und Server-Systeme festgelegt.

C Ergänzende Sicherheitsanalyse

In der ergänzenden Sicherheitsanalyse wird zunächst ermittelt, ob die Schutzbedarfe der IT-Anwendungen mit den BSI-Standardbausteinen abgedeckt werden können (Tabelle 4).

Komponente	**Begründung für Gefährdungsübersicht und Risikoanalyse**
A1: Webanwendung	Durch den Baustein B 5.21 und die dazugehörigen Maßnahmen können alle bekannten Gefährdungen abgedeckt werden.
A2: Webserver	Durch den Baustein B 5.4 und die dazugehörigen Maßnahmen können alle bekannten Gefährdungen abgedeckt werden.
A3: Webservice	Durch den Baustein B 5.24 und die dazugehörigen Maßnahmen können alle bekannten Gefährdungen abgedeckt werden.
A4: Datenbank	Durch den Baustein B 5.7 und die dazugehörigen Maßnahmen können alle bekannten Gefährdungen abgedeckt werden.

Tab. 4: Ergänzende Sicherheitsanalyse für IT-Anwendungen

Für jede IT-Anwendung gibt es einen passenden Baustein, und keine der Anwendungen weist einen sehr hohen Schutzbedarf auf. Hier muss keine Risikoanalyse durchgeführt werden. Dasselbe gilt für die Server-Software SSW1:Docker Software (Baustein B 1.10 und die dazugehörigen Maßnahmen). Für die Server-Systeme S1 und S2 ist dagegen eine Risikoanalyse durchzuführen, da die Technologie der Container nicht vollständig durch bestehende BSI-Bausteine abgedeckt werden.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 249

Ein integriertes Vorgehensmodell zur Planung und Umsetzung eines ISMS am Beispiel der Pharmaproduktion

Robert Geiger[1], Sabrina Krausz[2], Holger Mettler[3]

Abstract: Der Beitrag stellt ein integriertes Vorgehensmodell zur Planung und Umsetzung eines Informationssicherheitsmanagementsystems (ISMS) für KRITIS Betreiber im pharmazeutischen Produktionsumfeld vor. Es soll Betreibern kritischer Infrastrukturen helfen diese zu schützen und kann einen Beitrag zu einem branchenspezifischen Sicherheitsstandard (B3S) für den Sektor Gesundheit leisten. Es soll mögliche Synergien zu vorhandenen Systemen und Prozessen der pharmazeutischen Qualitätssicherung aufzeigen und zusätzliche Anforderungen der automatisierten Produktion berücksichtigen.

Keywords: IT-Sicherheitsgesetz, Informationssicherheitsmanagementsystem (ISMS), KRITIS Infrastrukturen, Risikomanagement, Risikobehandlung, Industrial Control Systems (ICS)

1 Einleitung

Die zunehmende Vernetzung und Anbindung von automatisierten Produktionsanlagen an komplexe IT-Systeme stellen neue Anforderungen an die Sicherheit von Maschinen und Anlagen. Informationstechnisch vernetzte Anlagen in und außerhalb der Produktion müssen daher hinsichtlich Cyber Security Anforderungen und Maschinensicherheit von den Betreibern und Hersteller von Automationslösungen neu überdacht werden. Hinzu kommen neue gesetzliche Anforderungen durch das IT-Sicherheitsgesetz, welches auch Teile der Pharmaindustrie verpflichtet den Stand der Technik umzusetzen um die kritischen Infrastrukturen zu schützen. Ein umfassendes Sicherheitskonzept betrifft in der Pharmaindustrie nicht nur die IT-Systeme sondern auch automatisierte, computerisierte Produktionssysteme. Da technische Sicherheit nur in Verbindung mit organisatorischen und personellen Maßnahmen wirkt, benötigt jede Organisation ein System von Verfahren, Prozeduren und Regeln zum Management der betrieblichen Informationssicherheit, ein ISMS. [Al16] Aus unserer Sicht ist es notwendig, beim Aufbau dieser Systeme die heterogenen Richtlinien und speziell, die oft schon vorhandenen technischen und organisatorischen Maßnahmen aus den (IT)–Qualitätssicherungsprozessen mit einzubeziehen. Insbesondere gängige Modelle zur Computer System Validierung (CSV) wie ISPE GAMP 5 [IS08] müssen hier integriert werden.

[1] M+W Central Europe GmbH, CSV & Cyber Security, Loewentorbogen 9b, 70376 Stuttgart, robert.geiger@mwgroup.net
[2] Hochschule Neu-Ulm, Wileystraße 1, 89231 Neu-Ulm, sabrina.krausz@student.hs-neu-ulm.de
[3] M+W Central Europe GmbH, CSV & Cyber Security, Loewentorbogen 9b, 70376 Stuttgart, holger.mettler@mwgroup.net

doi:10.18420/sicherheit2018_19

2 KRITIS - Absicherung der automatisierten Produktion

Zur Automation von Herstellungsprozessen und der Überwachung in der pharmazeutischen Produktion kommen industrielle Steuerungssysteme (engl. Industrial Control Systems - ICS) zum Einsatz. ICS haben neben den klassischen Schutzzielen Verfügbarkeit, Integrität und Vertraulichkeit noch andere Anforderungen. Dies äußert sich beispielsweise in längeren Betriebszeiten und seltenen Wartungsfenstern, z.B. für das Einspielen von Software-Updates oder -Patches. Zudem sind insbesondere die Echtzeitanforderungen zu nennen, die für die Steuerung häufig unerlässlich sind. Etablierte Schutzmaßnahmen aus dem Büroumfeld (z. B. Virenscanner) sind dabei nur bedingt auf ICS übertragbar. Auch wird der Lebenszyklus von ICS in der Regel aus dem Zyklus der zugehörigen Produktionsanlagen abgeleitet. Die Laufzeit, im Gegensatz zu typischer Office-IT, beträgt zehn bis sogar 20 Jahre. Echtzeit und Security sind also divergierende Anforderungen für ICS Systeme im Vergleich mit Office-Systemen.

2.1 ISO 27001 mit PDCA-Zyklus als Referenzprozess

Bis es einen KRITIS-konformen Standard für die Medizintechnik, Arzneimittelproduktion und weitere Unternehmen der pharmazeutischen Industrie gibt, kann der im IT-Sicherheitsgesetz geforderte Stand der Technik durch etablierte Methoden eines ISMS umgesetzt werden. Da die Implementierung nach ISO 27001 [DI17] Betreibern Kritischer Infrastrukturen empfohlen wird [KP16], wird im Modell diese Norm für den Referenzprozess genutzt. Auch die internationale Anerkennung und höhere Flexibilität sprechen für diese Norm. Die PDCA-Zyklen, die nicht nur implizit in der ISO 27001, sondern auch explizit integraler Bestandteil der Norm IEC 62443 [Ko16] sind, ermöglichen neben der Aufrechterhaltung und Verbesserung der Informationssicherheit auch die Möglichkeit der kontinuierlichen Überprüfung und Verbesserung der GMP-Praxis. Unter GMP (engl. Good Manufacturing Practice, dt. Gute Herstellungspraxis) versteht man internationale Richtlinien und Gesetze zum Qualitätsmanagement der Produktionsabläufe und -umgebung in der Produktion von Arzneimitteln und Wirkstoffen. [Eu11]

2.2 Integratives Vorgehen

Ausgangspunkt des integrierten Vorgehensmodells ist die Erstellung einer Cyber Sicherheitsstrategie speziell für automatisierte Produktionssysteme. Auf deren Basis können die oft schon zahlreich in der regulierten Pharmaindustrie vorhandenen IT Sicherheitsmaßnahmen, Leitlinien, Standard Operation Procedures (SOPs) und organisatorischen Maßnahmen für alle relevanten IT-Systeme abgefragt und berücksichtigt werden und an die Anforderungen eines modernen gesetzeskonformen ISMS angepasst werden. Unser Vorgehensmodell setzt die Anforderungen und Elemente der ISO 27001 mit den Maßnahmen der klassischen pharmazeutischen Qualitätssicherung und gesetzlichen Anforderungen (EU-GMP Leitfaden Anhang 11 [Eu11], FDA 21 CFR Part 11), CSV gemäß ISPE GAMP 5 [IS08] und Infrastrukturqualifizierung mit technischen, sowie organisato-

rischen Sicherheitsmaßnahmen im Produktionsbereich (IEC 62443) in Beziehung, siehe Abb. 1. Die Grundlage bildet ein Modell, das sich am ISPE GAMP Lebenszyklus orientiert und Überschneidungen der ISO 27001, GAMP 5 und IEC 62443 aufzeigt. [MG17]

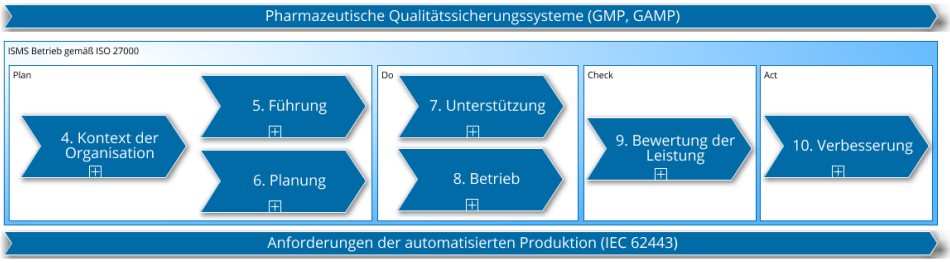


Abb. 1: Integratives Vorgehen am Referenzprozess der ISO 27001

2.3 Das Modell

Das kontrollflussorientierte, integrierte Vorgehensmodell soll die Planung und Umsetzung eines ISMS für KRITIS-Betreiber im Pharmaumfeld unterstützen. Es werden Prozesslandkarten und Unterprozesse auf Basis von Elementen der Business Process Modeling and Notation (BPMN) genutzt. Das Modell zeigt die Abfolge der Prozessschritte des Referenzprozesses aus ISO 27001, deren obligatorische und sinnvolle Dokumentation und die Zuordnung der Controls aus Anhang A, die direkt das ISMS betreffen. Überschneidungen zwischen Anforderungen der ISO 27001 und vorhandenen Systemen und Prozessen der pharmazeutischen Qualitätssicherung sind kenntlich gemacht. Diese werden bei der Implementierung durch Fragebögen und Checklisten analysiert. Tätigkeiten, die in ISO 27001 gefordert werden, wie etwa die Risikoanalyse bzw. Risikobehandlung bauen dann auf vorhandenen GMP-Analysen und Bewertungen auf und können durch neue Sicherheitsanforderungen ergänzt werden. Überschneidungspunkte des Modells mit vorhandenen Prozessen und Dokumentationen sind unter anderem folgende Themenbereiche: Asset-Management, Sicherheitsrichtlinien und Sicherheitsstrategie, Gefährdungsübersichten, Festlegung des Schutzbedarfs von Assets, Konzeption und Implementierung eines Identitäts- und Zugriffs-Managements, Sicherheitsvorfallmanagement, siehe auch Tab. 1.
Das integrierte Modell zeigt, dass durch die pharmazeutischen Qualitätssicherungsprozesse bereits eine starke Grundlage für die Einführung und Verwendung der Informationssicherheitsstandards ISO 27001 und IEC 62443 in der Pharmaindustrie besteht. Damit bietet sich für alle Unternehmen die nach diesen Richtlinien arbeiten wollen eine Möglichkeit kostengünstig und effizient Synergien auszunutzen.

Das integrierte Vorgehensmodell (Work-in-Progress), soll weiter entwickelt werden, mit der wissenschaftlichen Gemeinschaft und Partnern aus der Praxis diskutiert werden und innerhalb eines Industrieprojekts bewertet werden.

ISO 27001		In Pharmaindustrie vorhandene Prozesse/Dokumente
Kontext der Organisation	4.1	Sicherheits-Politik; IT-Sicherheits-SOPs; operative IT-Betriebsrichtlinien; IT-Betriebshandbücher; EHS (Environment-Health-System)
	4.2	bestehende Kontakte zu Zulassungsbehörde, Überwachungsbehörde, Aufsichtsbehörde
	4.3	Inventarisierungsliste; Systeminventarisierung; System Topologien; Site Master File und Validierungsmasterplan (EU-GMP-Leitfaden Kapitel 4 und Anhang 15); Zulassungsantrag; QM-Handbuch
Planung	6.1	Risikomanagement und -Bewertung nach EU-GMP Leitlinie Anhang 11; GAMP 5; ICH Q9
	6.2	IT-Sicherheitsrichtlinie, IT-Betriebsrichtlinie

Tab. 1: Beispiele für Synergien mit vorhandenen Prozessen/Dokumente der Pharmaindustrie

3 Literaturverzeichnis

[Al16] Altrhein, Adrian et.al.: Handreichung zum "Stand der Technik" im Sinne des IT-Sicherheitsgesetzes (ITSiG) (TeleTrusT - Bundesverband IT-Sicherheit e.V Hrsg.), Berlin, 2016.

[DI17] DIN-Normenausschuss Informationstechnik und Anwendungen (NIA) Informationstechnik – Sicherheitsverfahren – Informationssicherheitsmanagementsysteme – Anforderungen (ISO/IEC 27001:2013 einschließlich Cor 1:2014 und Cor 2:2015). Beuth Verlag GmbH, Berlin, 2017.

[Eu11] European Commission: EudraLex Volume 4 - EU Guidelines for Good Manufacturing Practice Medicinal Products for Human and Veterinary Use - Annex 11: Computerised Systems, Brussels, 2011.

[IS08] GAMP 5. A risk-based approach to compliant GxP computerized systems (German version). ISPE, Chicago Ill., 2008.

[Ko16] Kobes, P.: Leitfaden Industrial Security. IEC 62443 einfach erklärt. VDE Verlag, Berlin, 2016.

[KP16] Kipker, D.-K.; Pfeil, D.: IT-Sicherheitsgesetz in Theorie und Praxis. In Datenschutz und Datensicherheit - DuD, 2016, 40; S. 810–814.

[MG17] Mettler, H.; Geiger, R.: Sicherstellung der Cyber Security bei automatisierten Produktionssystemen. In Die Pharmazeutische Industrie, 2017, 79; S. 1164–1171.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 253

Fallstricke bei der Inhaltsanalyse von Mails: Beispiele, Ursachen und Lösungsmöglichkeiten

Steffen Ullrich[1]

Abstract: E-Mail ist eine der Hauptangriffswege zur Infektion mit Malware und zum Phishing von Zugangsdaten. Waren Mails vor 1996 auf ASCII-Zeichen und eine Zeilenlänge von 1000 Zeichen beschränkt, so ermöglicht die Nutzung der MIME-Standards heute die Abbildung beliebiger Zeichenkodierungen und binärer Anhänge innerhalb der ursprünglichen Beschränkungen. Die durch die Komplexität und Flexibilität dieser Standards bedingten Implementationsdifferenzen ermöglichen jedoch die Konstruktion von Mails, welche unterschiedlich in Sicherheits- und Endsystemen interpretiert werden. Wir haben exemplarisch untersucht, wie dadurch die Analyse in existenten Sicherheitsprodukten umgangen werden kann und welche Möglichkeiten es gibt, dieses Problem in der Praxis zu addressieren.

Keywords: Evasion, Semantic-Gap, Mail, MIME, Phishing, Malware, Firewall, IDS

1 Einführung

Die Abbildung der in heutigen E-Mails genutzten Features, wie beliebiger Zeichen und binärer Anhänge, auf die historischen Restriktionen von einer maximalen Zeilenlänge von 1000 Zeichen und nur ASCII, geschieht durch die Nutzung der MIME-Standards, welche 1996 und 1997 definiert wurden. Das folgende Beispiel zeigt eine typische Mail, in der die wichtigsten dieser Standards verwendet werden.

Deutlich erkennbar sind dabei die Definition von mehreren Mailteilen wie Text und Attachment mit jeweils eigenen Meta-Daten entsprechend RFC 2045 [FB96a] (Zeilen 7 bis 10 und 12 bis 15) und die Separation dieser Teile über einen textbasierten Trenner nach RFC 2046 [FB96b] (Zeilen 4, 6, 11 und 16). Da die Inhalte der Teile binäre bzw. nicht-ASCII-Zeichen haben, werden sie über die in RFC 2045 definierten Kodierungen Base64 (Zeile 15) bzw. Quoted-Printable (Zeile 10) zu ASCII-Text transformiert und dieses in den Meta-Daten entsprechend deklariert (Zeilen 8 und 13). Zusätzlich werden die Umlaute in den Meta-Daten der Mail bzw. Mailbestandteile kodiert, und zwar nach RFC 2047 [Mo96] im Subject der Mail (Zeile 3) sowie nach RFC 2231 [FM97] für den Dateinamen des Anhangs (Zeile 12):

[1] genua GmbH, 85551 Kirchheim bei München, Domagstr. 7, Deutschland, Steffen_Ullrich@genua.de

doi:10.18420/sicherheit2018_20

```
1  From: me@example.com
2  To: you@example.com
3  Subject: Viele =?UTF-8?Q?Gr=C3=BC=C3=9Fe?=
4  Content-type: multipart/mixed; boundary=trenner
5
6  --trenner
7  Content-type: text/plain; charset=UTF-8
8  Content-Transfer-Encoding: quoted-printable
9
10 Viele Gr=C3=BC=C3=9Fe von mir.
11 --trenner
12 Content-type: application/zip; name*=utf-8''%c3%bcbel.zip
13 Content-Transfer-Encoding: base64
14
15 UEsDBAoAAAAAAO6Tn0m2hH8nEwAAABMAAAAIA ...
16 --trenner--
```

Schon aus diesem Beispiel ist erkennbar, dass die Standards eine teilweise unnötige Flexibilität und damit einhergehende Komplexität aufweisen. Dazu kommt eine Mangel an klarer Definition in Grenzfällen. In der Praxis setzen daher viele Implementierungen nur einen Teil der Standards um und dieses oft leicht unterschiedlich. Zusätzlich werden nicht-standardisierte Erweiterungen eingesetzt. Durch gezieltes Ausnutzen dieser Implementationsdifferenzen kann ein Angreifer eine bösartige Mail so kodieren, dass sie in der gewünschten Weise vom Mail-Programm des Empfängers interpretiert wird, jedoch Mail-Filter, Antivirus-Produkte, Firewalls oder Intrusion-Detection-Systeme den Angriff nicht entdecken. Dieses Problem ist besonders brisant, weil Mail derzeit der primäre Verbreitungsweg für Malware und Phishing von Zugangsdaten ist. Wir konnten auch bereits Mails in der Praxis beobachten, die versuchen, derartige Interpretationsdifferenzen auszunutzen.

Im Folgenden zeigen wir exemplarisch einige Stellen, wo unterschiedliche Interpretationen typischerweise existieren. Wir haben untersucht, wie ein Analysesystem zur erfolgreichen Bekämpfung dieser Gefahren beschaffen sein muss und wie aktuelle Technologien und Produkte diese Anforderungen erfüllen.

2 Umgehung durch ambivalente und widersprüchliche Aussagen

Innerhalb der Meta-Daten wird zum einen die Kodierung des jeweiligen Mailteils deklariert (`Content-Transfer-Encoding`), wie auch die Art des Teils (`Content-Type`) sowie der textuelle Trenner bei Multipart-Mails (`boundary`). Auch wenn offensichtlich nur jeweils höchstens eine Deklaration Sinn macht, ermöglicht der Syntax der Meta-Daten mehrfache und widersprüchliche Deklarationen. Da die MIME-Standards keinen klaren Umgang mit solchen Fällen definieren, werden diese von Implementationen unterschiedlich behandelt. Als Beispiel zeigen wir das Verhalten bei einer widersprüchlichen Deklaration des Trenners in Multipart-Mails:

```
1  Content-type: multipart/mixed; boundary=foo
2  Content-type: multipart/mixed; boundary=bar
3
4  --foo
5  Content-type: text/plain
6
7  --bar
8  Content-type: application/octet-stream; name=malware.doc
7  Content-Transfer-Encoding: base64
8  ...
```

Hier benutzen die Mail-Clients Outlook und mutt die letzte Deklaration mit dem Trenner "bar". Sie ignorieren daher alles bis zum ersten Vorkommen von `--bar` in Zeile 7 als MIME-Preamble und sehen somit den Anhang mit dem Malware-Dokument. Apple-Mail, Thunderbird und Roundcube benutzen hingegen die erste Deklaration mit dem Trenner "foo" und interpretieren daher den Inhalt ab Zeile 7 als einfachen Text. Die gleiche Interpretation haben auch der Mail-Filter Amavisd, das Intrusion-Detection-System Snort und das Antivirus-Produkt ClamAV, d.h. diese interpretieren die Inhalte anders als das weit verbreitete Outlook. Ähnliche Effekte lassen sich mit einer Mehrfachdeklaration des `Content-Transfer-Encoding` erreichen. In dieser Situation benutzt Outlook nicht die letzte sondern die erste Deklaration und verhält sich damit anders als zum Beispiel das IDS Snort.

Auch die Kodierung von nicht-ASCII-Daten mittels Base64 oder Quoted-Printable selber kann verschieden interpretiert werden, zum Beispiel hinsichtlich Reaktion auf nicht erlaubte oder nicht erwartete Zeichen. So ended bei Base64 die Kodierung laut Standard eigentlich sobald ein Gleichheitszeichen auftritt. Demzufolge sollte im folgenden Beispiel Zeile 3 als Base64 interpretiert und Zeile 4 ignoriert werden. Die korrekte Dekodierung wäre damit "MALW" und wird so auch von Outlook oder mutt durchgeführt. Thunderbird und Roundcube hingegen machen mit der Dekodierung in Zeile 4 weiter und erhalten als Resultat entsprechend "MALWARE". Dieses Verhalten widerspricht den meisten Analysesystemen und ermöglicht so eine Umgehung der Analyse zum Beispiel bei Amavisd, Snort, ClamAV oder einer bekannten kommerziellen Firewall:

```
1  Content-Transfer-Encoding: base64
2
3  TUFMVw==
4  QVJF
```

Die bisher gezeigten Beispiele stellen nur einen kleinen Ausschnitt der gefundenen Interpretationsdifferenzen und damit einhergehenden Umgehungsmöglichkeiten dar. Weitere Varianten sind die Nutzung von nicht standardisierten Kodierungen wie uuencode oder y-encode, innovative Nutzung von Leerzeichen oder Zeilenumbrüchen an unerwarteten

Stellen oder einfach auch nur die Kodierung von binären Anhängen mit dem typischerweise für Texte genutzten Quoted-Printable statt Base64. Im Zuge der Forschungen haben wir eine umfangreiche Testsuite entwickelt, die einen stark automatisierten Test von Analysesystemen ermöglicht und so in kurzer Zeit eine Vielzahl von Problemen auch bei kommerziellen Systemen gefunden hat.

3 Ursachen und Lösungsansätze

Neben den bereits erwähnten tiefgreifenden Problemen der Standards selber, sehen wir ein mangelndes Verständnis bei den Autoren von Analysesystemen für diese Art von Umgehungen. Da zumeist davon ausgegangen wird, dass Mails eindeutig interpretierbar sind, werden Verfahren und Bibliotheken zur Extraktion der Inhalte genutzt, welche für Endsysteme nutzbar sind, aber nicht zur Analyse speziell präparierter Mails taugen.

Ein naiver Ansatz zum Umgang mit dem Problem wäre, alle potentiell ambivalenten Mails zu blockieren. Unsere Erfahrungen aus der Praxis zeigen jedoch, dass durch die große Vielfalt mailgenerierender Systeme derart problematische Mails in der Realität häufiger als gehofft auftreten. Analog legen wir auch keine Hoffnung in eine potentielle Überarbeitung der MIME-Standards, da hier die Vielfalt aller existenten Implementierungen berücksichtigt werden müsste und der neue Standard so noch komplexer und unbrauchbarer als bisher wird.

Die beste Option sehen wir in einer Konvertierung problematischer Mails in eine Form, bei der nur noch die von allen Systemen gleich verstandenen Kernfeatures der Standards benutzt werden. Da eine derartige Normalisierung die Modifikation der Daten bedeutet, kann sie nur in Systemen mit aktiver Analyse, wie Mail-Filtern und Applikations-Layer-Firewalls eingesetzt werden. Systeme mit passiver Analyse, wie Antivirus-Produkte, Intrusion-Detection-Systeme oder paketinspizierende Firewalls, müssten stattdessen sämtliche mögliche Interpretationsvarianten durchprobieren. Dieses scheitert in der Praxis aber bereits daran, dass nicht alle Varianten bekannt sind.

Literaturverzeichnis

[FB96a] Freed, Ned; Borenstein, Nathaniel S.: , Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Internet Requests for Comments, November 1996. http://www.rfc-editor.org/rfc/rfc2045.txt.

[FB96b] Freed, Ned; Borenstein, Nathaniel S.: , Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Internet Requests for Comments, November 1996. http://www.rfc-editor.org/rfc/rfc2046.txt.

[FM97] Freed, N.; Moore, K.: , MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations. Internet Requests for Comments, November 1997.

[Mo96] Moore, Keith: , MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text. Internet Requests for Comments, November 1996. http://www.rfc-editor.org/rfc/rfc2047.txt.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 257

Introducing DINGfest: An architecture for next generation SIEM systems

Florian Menges[1] Fabian Böhm[1] Manfred Vielberth[1] Alexander Puchta[2] Benjamin Taubmann[3] Noëlle Rakotondravony[3] Tobias Latzo[4]

Abstract: Isolated and easily protectable IT systems have developed into fragile and complex structures over the past years. These systems host manifold, flexible and highly connected applications, mainly in virtual environments. To ensure protection of those infrastructures, Security Incident and Event Management (SIEM) systems have been deployed. Such systems, however, suffer from many shortcomings such as lack of mechanisms for forensic readiness. In this extended abstract, we identify these shortcomings and propose an architecture which addresses them. It is developed within the DINGfest project for which we seek feedback from the community.

Keywords: Forensics; Virtual Machine Introspection; Visual Analytics; Security Incident and Event Management; Identity and Access Management

1 Motivation and Problem Statement

During the last few years, IT infrastructures have evolved to heterogeneous and complex structures which are becoming increasingly harder to control and protect. To reduce the complexity of securing those systems, companies have integrated centralized and holistic protection and incident management measures like Security Information and Event Management (SIEM) systems [Sh16]. SIEM systems mainly cover log collection, normalization, analyses, storage and monitoring [Mi11]. However, today's SIEM systems come with a lot of deficiencies which we address in the DINGfest (DetektIon, VisualisieruNG, ForEnsische Aufbereitung von SicherheiTsvorfällen) project for which we seek feedback from the community.

First, SIEM systems are quite expensive and proprietary. In DINGfest, we plan to show that it is possible to build an effective SIEM system based on open source software. This lowers the barriers for practical usage, especially for small and medium-sized businesses. Second, common SIEM systems are not prepared for forensic analysis, which is required if a legal

[1] University of Regensburg, Chair of Information Systems, surname.name@ur.de
[2] Nexis GmbH, alexander.puchta@nexis-secure.com
[3] University of Passau, Assistant Professorship of Security in Information Systems, (bt|nr)@sec.uni-passau.de
[4] Friedrich-Alexander University Erlangen-Nürnberg, Department of Computer Science, Security Research Group tobias.latzo@fau.de

doi:10.18420/sicherheit2018_21

dispute is expected. To improve the trustworthy collection of evidence, data is acquired via Virtual Machine Introspection (VMI) and the integrity of extracted data is protected to support a clean chain of custody. Third, DINGfest has built-in common forensic tools for an extensive and fast forensic investigation and will be able to normalize gathered evidences and report incidents to third parties, e.g., the German Federal Office for Information Security. In addition to common SIEM systems, we use further data sources for analysis, e.g., from an Identity and Access Management (IAM) system. Furthermore, users spend an extensive amount of time to configure detection heuristics of SIEM systems. In DINGfest we provide Visual Security Analytics that aim to combine automated analysis with domain knowledge of security experts. In the following we introduce the more general DINGfest architecture to overcome these shortcomings.

2 Conceptual Architecture

The DINGfest architecture consists of three main components (see Fig. 1). **Data Acquisition** collects data in real time from various sources. **Data Stream** represents the central data hub used by the subsequent component to query data, which is implemented using Apache Kafka. **Data Analysis** leverages event processing and identity behavior analytics (IBA) for automated incident detection. Additionally visual security analytics for integrating expert knowledge is incorporated. Finally, **Digital Forensics & Incident Reporting** transforms identified incidents into a structured format and reports them. The main components of the proposed architecture are currently under active development in the ongoing project.

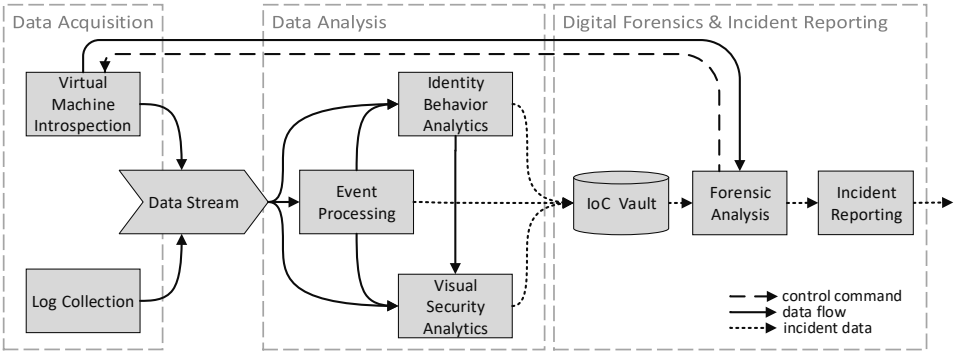


Fig. 1: DINGfest conceptual architecture and data flow

Data Acquisition: **Virtual machine introspection** (VMI) allows the acquisition of an untampered view on the system state from the outside without using an in-guest agent. Due to the strong isolation between the monitoring system and the monitored system, VMI is a valuable method for intrusion detection systems as well as for forensic analysis [Ja14]. Since VMI is resource intensive, we only gather a limited amount of information for intrusion detection. In the proposed architecture, we use VMI to periodically extract information

such as the process list of a monitored VM and trace system and function calls. In case of an incident, we collect additional information for forensic analysis. Besides that, we feed all system logs into the DINGfest data stream using the **log collection** module. Our VMI component is based on the libvmi library and supports the Xen hypervisor.

Data Analysis: The analysis within DINGfest is divided into different modules namely an event processing module, an identity behavior analytics module and a visual security analytics module. DINGfest's **event processing** module monitors the data stream, which consists of different sources like log data or system call traces. It recognizes the occurrence of pre-defined events using an event fingerprint database. Thereby, not only malicious events are detected but also benign ones. Afterwards, these events are passed to the following modules for further processing. To extend the event database, new fingerprints are computed based on the multiple execution of pre-defined events.
If data is generated via user actions (e.g. in applications) an additional **identity behavior analytics** (IBA) module can be applied. The term IBA was developed during the work of DINGfest and is based on the field of behavior analysis [SSW16]. Within this module, suspicious data packets can be identified based on the usage behavior of the underlying user permissions. Traditionally, IAM systems can be found in medium to large-sized organizations containing various information about users as well as their permissions and other attributes (e.g. department) [Hu16]. This data can be exploited for further analysis in the IBA module. Therefore, DINGfest maps data packets to an identity and its corresponding permission set. Thus, the module can compare the current behavior of users and their permissions with already known behavior (e.g. based on peer-group or history analysis).
The **visual security analytics** module combines automated analyses with interactive visualizations for an effective reasoning on large and complex data sets [Ke08]. In DINGfest, visual analytics is applied to include the domain knowledge of security experts into the analysis processes. Therefore, the two main analysis steps (event processing and identity behavior analytics) have to be supported by interactive knowledge-assisted visual interfaces. A central challenge here is to identify appropriate visual metaphors which allow appropriate conversion of knowledge between human and machine. This enables the continuous integration of expert knowledge and improves the detection rate of automated incident detection. Besides the support of the two analysis steps, DINGfest also provides an interface to visualize detected security incidents. This improves awareness for experts on the security situation, generates trust in the automated detection mechanisms, and allows to understand the incident. These components are based on open-source technologies (Angular 5, D3.js).

Digital Forensics & Incident Reporting: This component is responsible for preserving evidence that has been detected by the DINGfest analysis and to prepare this data to be reported to governmental institutions. The detected data is converted into Indicators of Compromise (IoC), which represent indications for incidents in a structured manner. These are stored into a graph database (**IoC vault**) which serves as information source for the

modules forensic analysis and incident reporting.
DINGfest's **forensic analysis** comes with an event fingerprint database which stores a large number of malicious and benign event fingerprints (as described above). Since only unique patterns are part of a fingerprint, the detection of a specific event can be considered as "forensically clean". To support the process of a chain of custody, extracted data is signed and hashed with the corresponding timestamp. For more elaborate forensic analysis, the hypervisor is extended with forensic tools. That allows the use of classic open source forensic analysis software.
The **incident reporting** module is responsible for reporting detected threats, attacks, and vulnerabilities. As soon as new incidents are identified and all relevant information is preserved by the forensics module. The information will be unified and transferred into a structured reporting format. It provides reporting capabilities for incident data to governmental institutions being compliant with applicable laws and directives such as the General Data Protection Regulation. Additionally, it enables the exchange of data within a cyber threat intelligence community, to quickly inform all participants about approaching threats. The module also provides pseudonymization capabilities to protect the reporting company's identity. In addition, procedures to create incentives for the exchange of threat information are researched and developed to increase the acceptance amongst companies.

Acknowledgment This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de). We wish to thank Günther Pernul, Hans Reiser and Felix Freiling for comments on an earlier version of this extended abstract.

References

[Hu16] Hummer, Matthias; Kunz, Michael; Netter, Michael; Fuchs, Ludwig; Pernul, Günther: Adaptive identity and access management—contextual data based policies. EURASIP Journal on Information Security, 2016(1):19, 2016.

[Ja14] Jain, B.; Baig, M. B.; Zhang, D.; Porter, D. E.; Sion, R.: SoK: Introspections on Trust and the Semantic Gap. In: 2014 IEEE Symposium on Security and Privacy. pp. 605–620, May 2014.

[Ke08] Keim, Daniel A.; Andrienko, Gennady L.; Fekete, Jean-Daniel; Görg, Carsten; Kohlhammer, Jörn; Melançon, Guy: Visual Analytics: Definition, Process, and Challenges. In: Information Visualization - Human-Centered Issues and Perspectives, volume 4950 of Lecture Notes in Computer Science, pp. 154–175. Springer, 2008.

[Mi11] Miller, David; Harris, Shon; Harper, Allen; VanDyke, Stephen; Blask, Chris: Security information and event management (SIEM) implementation. Network pro library. McGraw-Hill, New York, NY, 2011.

[Sh16] Shackleford, Dave: , SANS 2016 Security Analytics Survey, 2016. https://www.sans.org/reading-room/whitepapers/analyst/2016-security-analytics-survey-37467.

[SSW16] Shashanka, Madhu; Shen, Min-Yi; Wang, Jisheng: User and entity behavior analytics for enterprise security. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE, pp. 1867–1874, 2016.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 261

My Data is Mine - Users' Handling of Personal Data in Everyday Life

Sven Bock[1]

Abstract: This experimental study is about investigating users' handling of personal data and their awareness of data collection. A deception experiment was designed to let the subjects believe that they are participating in a decision-making experiment. Only after the experiment, they were informed about the actual aim of examining their behaviour towards their personal data. Before the deception experiment either a printed or a digital version of the terms and conditions was handed out. The reading time and the willingness to accept the terms and conditions were measured in order to find significant differences. For the deception, a program was implemented which simultaneously presents two terms including sensitive data like religious and political orientations. The subject should choose the favoured term. Afterwards, subjects were asked whether and to what extent they agree to hand out their collected data to third parties in exchange for financial gain. After the experiment the participants were asked about their usual behaviour regarding their personal data.

Keywords: Terms and conditions, Privacy, Transparency, End user license agreement, Personal data, Data collection.

1 Introduction

"I accept". These are the words you get confronted with at the end of each online transaction. It does not matter if it is the purchase of a physical item or an application. In any case the agreement is demanded by the declaration of consent. Often those terms are extensive and complicated written in legal language, so that very few users understand them. End users are usually unaware of the extent to which their data and rights are being used and the consequences of having them handed out. Especially nowadays in the era of technology, in which 86% of all households own a computer 58% of the german-speaking population a smartphone and 26% of them a tablet, the subject of data protection and data security should be highly valued.

Companies spend a lot effort collecting and analysing users' data. The high value of the data could be explained by the fact that the market is becoming more and more saturated and the companies strive for the one-to-one marketing with additional information about the costumer behaviour [Ba00]. Various concepts and techniques are increasing the efficiency and enable targeted marketing strategies and sales programs. The goal is to influence the people's consumer behaviour so that they acquire products promoted by the company. In addition to consumer behaviour, personality profiles are created which, among other things, allow an assessment of the creditworthiness. A survey in the US found that

doi:10.18420/sicherheit2018_22

75% of the population believe they have lost control over their personal information and thus are seeing themselves as “glass consumers” which are vulnerable to manipulations by companies and felt that companies manage too much personal information. A similar attitude is shown regarding privacy toward state authorities, as for an example a survey in Germany showed [Ku]. With an increasing amount of data and analysis possibilities in the hand of companies and state authorities, an increasing amount of monitoring and control of individuals is possible. This development is boosted by global players like Alphabet Inc., Amazon and Facebook which are omnipresent. It is assumed that the ownership of a mobile phone is always associated with the disclosure of information to large corporations which allows them to create a detailed profile of the customers. Goal of this empirical study is to examine if people are aware of handing out their data and how serious this problem is perceived. In the next step a solution should be found to make people more sensible with handing out private data.

2 Current state of research

The time when mobile phones were only used for verbal communication is long gone. Because of the faster mobile data connection smartphones and tablets are preferably used to receive information and services, such using emails, social networks, financial and health services, and other services [St]. This results in an increasing number of applications on the market offering various services. It is questionable how safe it is for someone to move through this mobile and digital world. Previous incidents have revealed numerous vulnerabilities in commonly known applications [Ki] as well as attacks from malicious software. Therefore, a secure mobile environment is currently not given. There is a large number of risks in the field of mobile devices, which concerns malware as well as the theft of confidential information and the reading of text messages. Thus, data security and privacy are a major concern for businesses and mobile end users [JS12]. Despite the popularity of smartphones, there are reasons to believe that due to concerns about security and privacy, the full potential of mobile devices is not being exploited by the users. A recent study found that 60% of smartphone users are concerned about financial and personal security risks in mobile payments [Ch12].

While end users often claim that data privacy is important for them, their actual behaviour towards sharing data and rights or installing applications is permissive. However, some studies show that privacy plays a relevant role in the installation decision. Users would always choose the applications with better privacy policies if functionality is not compromised [Go05]. The difference between the intention to protect private data and the actual permissive behaviour of individuals is known as the Private Paradox. This paradox is often explained with the Private Calculus, meaning that individuals perform a cost-benefit calculation in a situation where they can disclosure private data in order to gain compensation. This approach has been expanded by Acquisti and others to include factors such as incomplete information and the desire for instant gratification. In doing so it was transformed from a rational based explanation towards an explanation based on behavioural science [ABL15].

Terms and conditions are clear indicators for security and privacy level. Few users pay attention to these agreements before installing an application. Although, surprisingly, 60% of end users say that they consider the shared rights "sometimes" or "always". Even though, the shared rights are classified as fairly unimportant. These findings suggest that end users rely more on reviews and recommendations than on the difficult-to-understand conditions of security and privacy [Ch12]. In general, end users are unaware of the serious impacts of confirming a dialogue box in economic, social and legal terms. Considerable attention and cognitive effort would be needed to respond appropriately to those dialogues. Both are often absent because the user usually gets into this situation while he wants to do another primary task which has to be interrupted.

3 Research Question

Goal of this study is to determine if people are aware of handing out their data and how serious this problem is perceived. Furthermore the thesis is eliciting the value people assign to their private data. The study focuses on the field of mobile devices, because of their increasing importance. Following hypothesis were postulated:

H1: Although users claim that data protection is important to them, their actual indifferent behaviour regarding data protection is caused by undiscerned consequences. There are three distinguishable parts in this hypothesis:
H 1.1: Users report that it is important to protect their data but do not behave accordingly.
H 1.2: Users are not aware of the consequences while handing out their data and rights.
H 1.3: Their indifferent behaviour regarding data protection is caused by undiscerned consequences.

Hypothesis 1.1: This hypothesis is based on the privacy paradox which claims that intend and behaviour regarding data privacy differ. On one side people were revealing that data security and data protection are highly relevant for them and for their choice of online shops. Especially the transparency of the terms and conditions has a high importance. On the other side they are permissively handing out their data for getting a better service [Kö15].

Hypothesis 1.2: Studies showed that end-users do not read contracts. This phenomenon extends from paper-contracts to CTAs [BMWT09]. Plaut and Bartlett examined individual reasons for not reading contracts, among them are length and complexity of the contract [PBI12]. Based on this behaviour it can be assumed that end-users are uncertain about the consequences of handing out personal data.

Hypothesis 1.3: This hypothesis builds a causal connection between H 1.1. and H 1.2.: The indifferent behaviour is caused by being unaware of the consequences of handing of private data. This is based on Mischels delay-of-gratification paradigm which shows that people tend to favour short-term over long-term benefits even though the long-term benefits are more substantial [Mi10]. In this case the short-term benefit is the functions of an application while the long-term benefit is the maintenance of privacy.

H2: The users are handing out their information for a low compensation.

Hypothesis 2: A study of Staiano et. al. showed that people are willing to sell their data for a low compensation regardless of their socio financial background [St14].

4 Experimental Setup and Design

An experiment was designed consisting of six steps which are illustrated in Fig. 1. One important consideration during the development was the response behaviour of the subjects regarding data protection which was assumed to be given according to social expectations. Regarding this, a deception experiment setup has been chosen as the goal was to examine the actual behaviour and not the pretended opinion. The subjects were made to believe that they are participating in a decision-making experiment. Actually, a closer look has been taken on the behaviour towards the release of personal data in exchange of financial offsets.

In the decision-making experiment the user had to choose between two opposing terms for 450 times. The chosen terms included private issues such as political and religious orientation. In this phase of the experiment the focus was put on the users' behaviour towards the agreement procedure of the terms and conditions to participate on the experiment. The terms of this agreement were designed exaggerated, so while reading the subjects could be aware that they were handing out all their rights and information. One example is that the terms and conditions included a permission to hand out personal data without any beforehand approval. The sample was divided into two groups. One group got the agreement presented in paper form while the other group got it in digital form as a click-through agreement. For both groups the users' reading time for the agreement was measured.

After completing the decision-making experiment, the subjects were told that their data would be of interest to other institutions as well. Six groups of universities and companies were presented to them. The subject should decide whether and to whom the data should be passed. For choosing not to pass the data at all the lowest financial compensation was promised. Users were offered a higher financial compensation if they choose to share their data with more institutions. This was followed by a short interview including seven questions for examining if the testee noticed something strange and aiming to get subjective impressions about handing out the data. The interview was followed by a questionnaire with 22 questions, inquiring the subjects' usual behavior while installing new mobile applications. Furthermore the participants were asked about which data they think is handed out by the application and the assumed financial value of their personal data. In addition to the present experiment, the questionnaire was also used as an online survey. Aim is to examine if there is a significant difference between the two samples, especially since the people that participated in the experiment had just been in a situation in which they had been confronted with their handling of private data.

At the end it was revealed to the participants that they took part in a deception experiment. Also all participants received the same financial compensation, regardless of their choice of data transfer. The experimental design was accepted by the Ethics committee

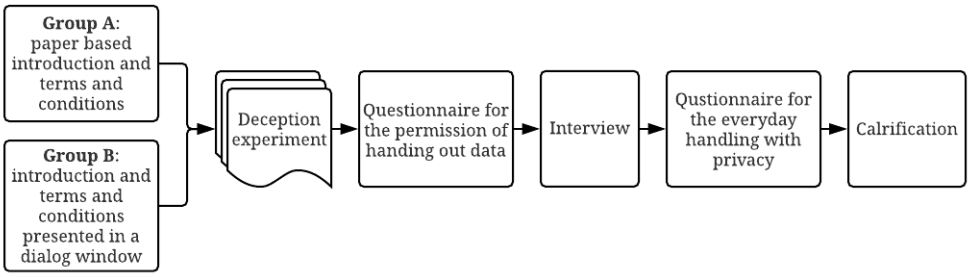


Fig. 1: Chronological process of the study.

after implementing some improvements. 51 participants were recruited using the online test person portal from the Technical University of Berlin, from which 20 of them were male. The average age was 31.27 years (SD 10.22; range: 18-64 years).

5 Results

While a thorough analysis of the data is still ongoing, a first descriptive evaluation shows that all participants (N = 51) signed the agreement. The mean reading time for the agreement in paper form was: M = 58.29s, SD = 53.25s and for the digital form: M = 51.79s, SD = 52.47s. The agreement contained 833 words while the average reading speed is about 250 words per minute. This points out that the agreement has not been read completely regarding the average reading time.

On the other hand, in the questionnaire the participants pointed out that the protection of their private data is highly valued. On an eleven-staged rating scale (0 lowest, 10 highest value) more than 23% had chosen the highest value. Also 86% of the participants had chosen to share their data with the highest amount of institutions, for the highest financial gain, followed by 10% for the second highest compensation only 4% took the option not to share their data at all. At the end of the experiment, it was observed that the testees invested a lot more time in reading the clarification in contrast to the agreement before.

6 Conclusion

The rough analysis of the presented experiment has already indicated that people show a paradoxical behaviour regarding the value of their data and the actual treatment of them. This points out that the issue is a very interesting field of research and that the data of the experiment should be deeper analysed, especially regarding the increasing number of mobile device users. The next step is to find a solution for educating mobile device users and to make them more sensitive with handing out their data to prevent them of being transparent consumers.

References

[ABL15] Acquisti, Alessandro; Brandimarte, Laura; Loewenstein, George: Privacy and human behavior in the age of information. Science, 347(6221):509–514, 2015.

[Ba00] Baeriswyl, Bruno: Data Mining und Data Warehousing: Kundendaten als Ware oder geschütztes Gut. Recht der Datenverarbeitung, 1:6–11, 2000.

[BMWT09] Bakows, Y; Marotta-Wurgler, F; Trossen, DR: Does anyone read the fine print. A test of the informed minority hypothesis using clickstream data. New York University School of Law Working Paper, 2009.

[Ch12] Chin, Erika; Felt, Adrienne Porter; Sekar, Vyas; Wagner, David: Measuring user confidence in smartphone security and privacy. In: Proceedings of the eighth symposium on usable privacy and security. ACM, p. 1, 2012.

[Go05] Good, Nathaniel; Dhamija, Rachna; Grossklags, Jens; Thaw, David; Aronowitz, Steven; Mulligan, Deirdre; Konstan, Joseph: Stopping spyware at the gate: a user study of privacy, notice and spyware. In: Proceedings of the 2005 symposium on Usable privacy and security. ACM, pp. 43–52, 2005.

[JS12] Jain, Anurag Kumar; Shanbhag, Devendra: Addressing security and privacy risks in mobile applications. IT Professional, 14(5):28–33, 2012.

[Ki] King, R.: ViaForencis: Netflix, Foursquare Apps Leave Sensitive Data Vulnerable. ZDNet.com, 2011.

[Kö15] Köln, IFH: , Perspektiven für den Datenschutz in Europa aus der Sicht der Verbraucher und des (elektronischen) Handels, 2015.

[Ku] Kurz, Constanze: Eco - Verband der Internetwirtschaft e.V., Umfrage zur Vorratsdatenspeicherung im Juni 2015. https://netzpolitik.org/2015/umfragen-zur-vorratsdatenspeicherung-klare-mehrheit-gegen-wiedereinfuehrun Accessed: 25/11/2017.

[Mi10] Mischel, Walter; Ayduk, Ozlem; Berman, Marc G; Casey, BJ; Gotlib, Ian H; Jonides, John; Kross, Ethan; Teslovich, Theresa; Wilson, Nicole L; Zayas, Vivian et al.: 'Willpower' over the life span: decomposing self-regulation. Social cognitive and affective neuroscience, 6(2):252–256, 2010.

[PBI12] Plaut, Victoria C; Bartlett III, Robert P: Blind consent? A social psychological investigation of non-readership of click-through agreements. Law and human behavior, 36(4):293, 2012.

[St] Statistisches Bundesamt. Anteil der privaten Haushalte in Deutschland mit einem Computer im Zeitraum 1998 bis 2015 (Stand: 1. Quartal 2015). http://de.statista.com/statistik/daten/studie/2596/umfrage/ausstattungsgrad-privater-haushalte-mit-einem-pc-seit-1998/. Accessed: 25/11/2017.

[St14] Staiano, Jacopo; Oliver, Nuria; Lepri, Bruno; de Oliveira, Rodrigo; Caraviello, Michele; Sebe, Nicu: Money walks: a human-centric study on the economics of personal mobile data. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, pp. 583–594, 2014.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 267

Bounded Privacy: Formalising the Trade-Off Between Privacy and Quality of Service

Lukas Hartmann[1]

Abstract: Many services and applications require users to provide a certain amount of information about themselves in order to receive an acceptable quality of service (QoS). Exemplary areas of use are location based services like route planning or the reporting of security incidents for critical infrastructure. Users putting emphasis on their privacy, for example through anonymization, therefore usually suffer from a loss of QoS. Some services however, may not even be feasible above a certain threshold of anonymization, resulting in unacceptable service quality. Hence, there need to be restrictions on the applied level of anonymization. To prevent the QoS from dropping below an unacceptable threshold, we introduce the concept of *Bounded Privacy*, a generic model to describe situations in which the achievable level of privacy is bounded by its relation to the service quality. We furthermore propose an approach to derive the optimal level of privacy for both discrete and continuous data.

Keywords: Privacy; Quality of Service; Modelling Anonymity

1 Introduction

In the last years, methods to collect and process personal data have been dramatically improved and are widely used nowadays. Some of these systems could be used for tracking individuals or to obtain more personal information by aggregating existing data. As a consequence, some users wish to reveal as little personal data as possible and to stay private. One widely used approach is the anonymization of sensitive data so that the reported data cannot be easily mapped to a specific subject. For some services however, exactly this sensitive data may be actually necessary for operation.

Location information can be highly sensitive, revealing not only specific locations like home or work, but also for example religious or political views. By only knowing the home and work location of a subject, one can reveal the identity of an anonymous individual with a very high probability [GP09]. If location information is tracked over time, one can infer even more personal data of this user. Therefore, *Location Privacy* is very important for privacy-conscious users. Location Privacy is defined as a state when the location of a subject is not revealed to other subjects. A widely-used approach is the obfuscation of the exact geographical location and consequent reporting of the obfuscated information.

[1] Universität Regensburg, Lehrstuhl für Wirtschaftsinformatik IV, 93040 Regensburg, lukas.hartmann@ur.de

doi:10.18420/sicherheit2018_23

Nevertheless, location based services (LBS) require location data to work properly and they require a certain precision in this data to provide their service with an acceptable level of quality. For mobile route planning with a feasible quality, a precise geographical information for start and target location has to be submitted by the user to the LBS. Reporting highly anonymized geo-coordinates would not be sufficent and would result in a unacceptable service quality. The routing app would be useless. On the contrary, there are apps for which a rough location information would be sufficient. A weather app only needs the city or at most the postal code, but no precise address information. When using LBS, there is always a trade-off between the quality of service (QoS) and the amount of location privacy a user can obtain. Depending on the individual service and use case, the user has to accept less privacy, if appropriate quality should not fall below a certain threshold.

Stipulated by German law, operators of critical infrastructure in Germany have to report security incidents to the Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik, BSI). Critical infrastructure in this context is divided in different industrial sectors covering "classicalïnfrastructure like transportation, energy and water, but also IT infrastructure, cultural sites and media. The law permits however that incidents can be reported with related data partially anonymized, so that the operator does not have to reveal too much information about the affected business. This anonymization could be done by for example by obfuscation of the geographical location, the time when the issue arose or by a generalization of the infrastructure category. In any case, the reporting should still contain enough information so that the incident can be treated by the BSI in an appropriate way. If a coal power plant might suffer from a coal shortage due to issues with the internal enterprise resource planning system, it is possibly sufficient to report the affected region with a certain precision. However, if an IT incident leads to a critical damage to a power plant, more precise information of the actual power plant is needed. In the scenario of incident reporting, there is always a trade-off between the quality of reporting and the amount of anonymization/privacy one can obtain. Depending on the actual incident, one has to use less anonymization in order to provide enough information in the incident report.

In both scenarios we have a trade-off between anonymization and the quality of service, where the achievable level of privacy is bounded by its relation to the service quality. To prevent the QoS from dropping below an unacceptable threshold, we introduce the concept of *Bounded Privacy* describing situations when the quality of anonymization is bounded by an upper bound. We present a generic model to this problem and introduce methods how to obtain Bounded Privacy depending on the given information. To the best of our knowledge, this is the first paper which investigates privacy when the corresponding service shall have a decent, pre-defined minimum quality.

2 Related Work

Several methods have been proposed in the literature to anonymize location information. The probably best discussed concept is the notation of *k-Anonymity* introduced

by Sweeney [Sw02]. An extension of this model called *l-Diversity* was introduced by Machanavajjhala et al. [Ma07]. In a similar vein, Ağır et al. [Ağ16] try to mask semantic information about visited locations by generalizing it along a hierarchical semantic tree. Andrés et al. [An13] propose the idea of *ε-Geo-Indistinguishability* which attempts to restrict the information leakage to an observing adversary by employing a perturbation mechanism that obfuscates real location information in a probabilistic way. This approach is based on the concept of *Differential Privacy* introduced by Dwork [Dw11], originally proposed for statistical databases.

Literature shows a great variation in exchange formats used for the reporting of IT security incidents. The most recent overview on this topic was presented by Menges and Pernul [MP18] focusing on the on applicability of exchange formats for IT security incidents.

3 Bounded Privacy

As stated in the introduction, there is often a trade-off between privacy and the quality of service. This leads to scenarios where a decent service quality has to be guaranteed and thus the anonymization level has to be bounded. In this chapter, we introduce the approach of *Bounded Privacy* leading to a feasible level of anonymization, that allows for the given threshold on the service quality.

In general, an information consists of geographical coordinates and semantic information, for example location tags, infrastructure categories, etc. Therefore we model the information I as an element of the *domain*

$$\mathcal{D} = \mathbb{R}^2 \times \mathcal{S},$$

where the geographical information is given as a pair $(x, y) \in \mathbb{R}^2$ of coordinates and the semantic information $s \in \mathcal{S}$ comes from a generic set. In most scenarios, $\mathcal{S}$ would be a multi-dimensional space where each dimension would represent one semantic attribute. A semantic information $s \in \mathcal{S}$ would be of the form $s = (s_1, s_2, \ldots)$ where s_1 could for example be a category (“energy“), s_2 could be an impact category (“high impact“), etc.

When we anonymize an information to obtain privacy, we apply an *anonymization function*

$$AN\colon \mathcal{D} \to \mathcal{P}(\mathcal{D}),\ I \mapsto AN(I)$$

to the original information I, consisting of the functions $AN_{geo}\colon \mathcal{D} \to \mathcal{P}(\mathbb{R}^2)$ and $AN_{sem}\colon \mathcal{D} \to \mathcal{P}(\mathcal{S})$ for obfuscating geographical and semantic information, respectively. The anonymized or obfuscated information $AN(I)$ is an element of the power set $\mathcal{P}(\mathcal{D})$ and hence a subset of the information domain $\mathcal{D}$, since common anonymization techniques use cloaking mechanisms to obtain privacy. The most popular example for this is k-Anonymity (see section 2).

For measuring how much two (possibly anonymized) pieces of information differ from each other, we introduce a generic *information distance* which is given by

$$d\colon \mathcal{P}(\mathcal{D}) \times \mathcal{P}(\mathcal{D}) \to \mathbb{R}^2_{\geq 0}.$$

It calculates both the geographical and semantic difference as non-negative numerical values. Analogous to the anonymization function AN, the information distance d consists of two functions d_{geo} and d_{sem} to measure the geographical and semantic distance, respectively.

The threshold below which the quality of service shall not fall - and as a consequence the bound on the level of privacy - is modelled via a *restriction function* r consisting of $r_{geo}, r_{sem}\colon \mathcal{D} \to \mathbb{R}$. With this function, we can define when an anonymization is valid with respect to the given threshold restrictions: An anonymization of a given information I is valid if

$$d(\{I\}, AN(I)) \leq r(I)$$

i.e. if both restrictions $d_{geo}(\{I\}, AN_{geo}(I)) \leq r_{geo}(I)$ and $d_{sem}(\{I\}, AN_{sem}(I)) \leq r_{sem}(I)$ on the information distance between original information I and anonymized information $AN(I)$ are fulfilled.

Depending on the use case and the individual information, one could have different restrictions which level of anonymization can be applied. As the level of allowed restrictions is dependent on the information quality necessary for the lower bound on the quality of service, its definition should be determined at least in part by the service provider. The restriction function r is also dependent on certain parts of the information I. Following the previously used example from section 1 comparing between a coal shortage and a nuclear melt-down, this means, that the restriction function $r(I)$ would have a smaller value in the latter case although both examples come from the same domain. One can see that the function r can have a significantly different figure within even a single scenario.

We call the concept *Bounded Privacy*, when the level of anonymization is bounded by a minimum level of service quality and therefore valid anonymization with respect to the given restriction function is necessary.

4 Methods for Obtaining Bounded Privacy

In the generic model from section 3, an information I consists of data from discrete and continuous domains. Geographic information is normally given by continuous geo-coordinates $(x, y) \in \mathbb{R}^2$, whereas semantic data is normally given as a tuple of discrete values, like categories, sectors or location tags.

To implement Bounded Privacy for discrete data, a bottom-up based tree approach could be used: The discrete information units represent the leafs of a tree. Related units can be grouped together and get a common parent node in the tree, representing the aggregated

information of its child nodes within a topical hierarchy. Instead of reporting the exact semantic information on the leaf level, one would report the aggregated information given by the parent nodes to anonymize the exact information. The further away from the leaves we move, the less detailed information is reported and therefore the anonymization is better. Following the example of critical infrastructure, categories like "Nuclear Power Plant" and "Coal Power Plant" could be leafs of the semantic tree and could be grouped together to the obfuscated category "Fossil Energy Power Plant". This parent category could be anonymized to the more general category "Power Plant" or even to "Energy". There are already efforts to create similarly structured semantic ontologies for different sectors that could be used as a basis for such topical hierarchies. This is an ongoing topic of research and could prove an important piece to bridge the gap between our theoretical model and practical applicability [SS10]. For measuring the information distance d, one could use the graph metric d_G which calculates the length of the shortest path between two given nodes. If a restriction $r_{sem}(I)$ is applicable for a given information I, it is only allowed to step that far in the semantic graph such that $d_G(I, AN_{sem}(I)) \leq r_{sem}(I)$.

For the anonymization of data derived from a continuous domain, an approach based on the concept of *ε-Geo-Indistinguishability* could be used. Originally, this mechanism was proposed for geographical data only, but one can adapt it for other domains of continuous data as well. The main idea of ε-Geo-Indistinguishability is the following: With a perturbation mechanism K, a geo-coordinate $(x, y) \in \mathbb{R}^2$ is mapped to another point (x^*, y^*) in a "probabilistic way". The obfuscated point will then be reported instead of the original location. Andrés et al. [An13] propose to use a two-dimensional Laplacian distribution which creates noise and obfuscates the geographical location. This mechanism can be used as anonymization function AN_{geo}. The distribution of the Laplacian noise highly depends on the parameter ε and influences the level of perturbation. Therefore, the restriction function r_{geo} has to restrict the distribution of the Laplacian noise so that the obfuscated points are not too far away from the original geographical location with a high probability. Since the anonymized geographical-information is given in probabilistic way, the information distance d_{geo} must also be measured probabilistically. A suitable approach for the restriction function r_{geo} would be to introduce a radius s around the original geo-coordinate (x, y) in which the obfuscated point (x^*, y^*) should be mapped with at least probability p. This probability would be dependent on the information I so that in some scenarios higher probabilities could be used as in others. In order to keep our model deterministic, we propose to redraw the obfuscated point (x^*, y^*), if the distance to the original point (x, y) is higher than allowed by our restriction function, to ensure that the QoS cannot drop below the specified bound.

5 Conclusion

We presented a generic model to the problem when privacy is bounded such that a minimum Quality of Service has to be guaranteed. We introduced generic anonymization functions and

tools for measuring and restricting the information distance between original and anonymized information. Furthermore, basic approaches on how to apply these concepts to discrete and continuous data were given. For the discrete domain, we used a tree-based approach in which a broader semantic tag/category shall be reported as long as this is valid with respect to the restriction function. A method which is based on ε-Geo-Indistinguishability was used for continuous geographical data. Noise was added to the original location following a planar Laplacian distribution.

In future work, it is envisioned to evaluate the generic model of *Bounded Privacy* on a real data example and define restriction functions for a concrete scenario like incident reporting. Furthermore, we want to extend the generic model so that upper bounds on the provided information are included as well. Another possible extension of the model would be the introduction of an evaluation mechanism for the reported information so that for example the monetary value of the anonymized information could be measured. With this approach, one could also investigate in which circumstances it might have a positive effect for a subject to report more information than needed, if this reporting has some benefits. Based on these two extensions, one could even quantitatively formalize the trade-off between privacy and QoS, thus enabling users to pick the optimal privacy level for their needs and ultimately realizing their informational self-determination.

References

[Ağ16] Ağır, Berker; Huguenin, Kévin; Hengartner, Urs; Hubaux, Jean-Pierre: On the Privacy Implications of Location Semantics. Proceedings on Privacy Enhancing Technologies, 2016(4):165–183, 2016.

[An13] Andrés, Miguel E; Bordenabe, Nicolás E; Chatzikokolakis, Konstantinos; Palamidessi, Catuscia: Geo-indistinguishability: Differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, pp. 901–914, 2013.

[Dw11] Dwork, Cynthia: Differential privacy. In: Encyclopedia of Cryptography and Security, pp. 338–340. Springer, 2011.

[GP09] Golle, Philippe; Partridge, Kurt: On the anonymity of home/work location pairs. In: International Conference on Pervasive Computing. Springer, pp. 390–397, 2009.

[Ma07] Machanavajjhala, Ashwin; Kifer, Daniel; Gehrke, Johannes; Venkitasubramaniam, Muthuramakrishnan: l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1):3, 2007.

[MP18] Menges, Florian; Pernul, Günther: A comparative analysis of incident reporting formats. Computers & Security, 73(Supplement C):87 – 101, 2018.

[SS10] Staab, Steffen; Studer, Rudi: Handbook on ontologies. Springer Science & Business Media, 2010.

[Sw02] Sweeney, Latanya: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557–570, 2002.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 273

Towards a Differential Privacy Theory for Edge-Labeled Directed Graphs

Jenni Reuben[1]

Abstract:

Increasingly, more and more information is represented as graphs such as social network data, financial transactions and semantic assertions in Semantic Web context. Mining such data about people for useful insights has enormous social and commercial benefits. However, the privacy of the individuals in datasets is a major concern. Hence, the challenge is to enable analyses over a dataset while preserving the privacy of the individuals in the dataset. Differential privacy is a privacy model that offers a rigorous definition of privacy, which says that from the released results of an analysis it is '*difficult*' to determine if an individual contributes to the results or not. The differential privacy model is extensively studied in the context of relational databases. Nevertheless, there has been growing interest in the adaptation of differential privacy to graph data. Previous research in applying differential privacy model to graphs focuses on unlabeled graphs. However, in many applications graphs consist of labeled edges, and the analyses can be more expressive, which now takes into account the labels. Thus, it would be of interest to study the adaptation of differential privacy to edge-labeled directed graphs. In this paper, we present our foundational work towards that aim. First we present three variant notions of an individual's information being/not being in the analyzed graph, which is the basis for formalizing the differential privacy guarantee. Next, we present our plan to study particular graph statistics using the differential privacy model, given the choice of the notion that represent the individual's information being/not being in the analyzed graph.

Keywords: Differential privacy, graphs, labels, analyze, utility.

1 Introduction

Analyzing the information collected in statistical databases provides valuable insights in the medical and social science research. However, the challenge is how to ensure that the public release of the results from the analysis does not compromise the privacy of the individual contributors of the dataset. This challenge is referred to as the data privacy problem and being extensively studied both in the statistics and computer science community [AW89, AS00, Be80, Ch05, Sw02].

Increasingly, more and more information is represented using graph structures, for example social network data, financial transactions and semantic assertions in the Semantic Web context. Many recent studies have investigated the data privacy problem in graph data [ZPL08].

[1] Department of Mathematics and Computer Science, Karlstad University, Sweden, jenni.reuben@kau.se

doi:10.18420/sicherheit2018_24

Among the studied privacy models, the Differential Privacy model provides a mathematical definition of data privacy that is guaranteed to the participants of a database, independent of the auxiliary information available to an adversary[3]. Intuitively, a trusted curator of the database uses a privacy preserving mechanism that satisfies the definition of differential privacy for releasing the results of the analysis performed on the database. The definition of differential privacy states that the result is 'essentially' the same when an individual participates or refrains from participating in the database. Thus, the publicly released results provide meaningful insights about the underlying population of the database yet obscure any one individual's contribution.

Previous research in applying differential privacy theory in the context of releasing graph statistics focuses on graphs with unlabeled edges. One typical example of the queries over the graph data is, '$COUNT$ all edges'. However, in many applications, the graphs consist of edges that are labeled. Accordingly, there are analyses that take into account these labels, for example to get '$COUNT$ of the edges that have certain label(s)' . It would thus be of interest to learn how to apply differential privacy to graphs with labeled edges, where different edges may have different labels. The main goal of this paper is to define the foundations for applying differential privacy to edge-labeled directed graphs.

2 Preliminaries

As a basis for presenting our work in this paper, in this section we recall the main notions defined for differential privacy both in the relational database settings [Dw08, Dw06] and in graphs without edge labels [Ka13, NRS07, TC12].

A database D is a set of rows. Consider a database D_I that contains information about a set of individuals I, where the information about each individual is captured as a separate row. Now, consider another database $D_J = D_{I \pm x}$ that includes/excludes information of one random individual x. So, D_J and D_I differ by one row and they are called neighboring databases [Dw08]. A trust worthy curator uses a privatized mechanism K that takes a database D as input and produces a result KD, which gives nearly zero evidence about whether the input database is D_I or D_J, thus obfuscating any one individual's contribution to the result.

Definition 1 (Differential Privacy [Dw08]) *A privatized mechanism K is said to give* ϵ-differential privacy*, if for any pair of databases D_I and D_J that differ by one row, and for all $S \subseteq RangeK$, it holds that: $PrKD_I \in S \leq e^{\epsilon} \times PrKD_J \in S$*

where the probabilities represent the random choices made by the privatized mechanism K and $RangeK$ denotes the set of all possible outputs of K.

[3] In this context, an adversary is an entity that intends to compromise the privacy of the participants of a database.

Dwork et al. [Dw06] presented a privatized mechanism K for continuous-valued queries, that is the classes of queries that map the database to vectors of real numbers. If the true response of a query function f is fD, for achieving ϵ-differential privacy the privatized mechanism then distorts this true response by adding appropriately chosen noise before disclosing it to the public. The noise that needs to be added to the true response is given by the sensitivity of the query function f and the chosen value of ϵ. The function sensitivity specifies what is the maximum difference that the privatized mechanism needs to bridge in the form of additive noise such that from the noisy response it is difficult to attribute that the input database is D_I or D_J. If the value of ϵ is set to a very small value, then the noise that need to be added increases. Similarly, the amount of the additive noise will be large if the sensitivity of the function is greater.

The original differential privacy definition remains essentially unchanged for graph data. However the notion of neighboring databases on which Definition 1 is based on need to be adapted to graph data. In the literature, there appear two variants of differential privacy definitions that formalize the privacy guarantee for two different notions of what it means for a pair of graphs to differ by one unit. One definition is *edge privacy*, which formalizes the differential privacy guarantee for any two graphs that differ by at most one edge [NRS07]. The second definition is *node privacy*, which deals with any pair of graphs that differ by a single node including all its adjacent edges [Ka13, TC12]. Inspired by these definitions, in the next section, we present three variants of differential privacy definitions that guarantee various levels of privacy protections for edge-labeled directed graphs.

3 Differential Privacy for Edge-labeled Directed Graphs

Let L be an infinite set of possible edge labels. An edge-labeled directed graph, hereafter simply a graph, is a tuple $G = V, E$, where V is a set of vertices, and E is a set of edges such that $E \subseteq V \times L \times V$. The privacy guarantee formalized in Definition 1 builds on the notion that the response for a query over a database is 'essentially' the same for any two databases that are neighbors (i.e they differ by a row). For graph data, as presented in Section 2 there are different possibilities to represent what it means for two graphs to differ by one unit. The following definitions specify what it means for a pair of graphs being neighbors for formalizing the differential privacy guarantee for edge-labeled directed graphs. Each possible definition of neighboring graphs provides a different semantic interpretations of the differential privacy guarantee. Hence it is important to study the privacy/utility trade-off of the chosen graph neighbor definition. First, we adapt the 'edge privacy' definition of unlabeled graphs. Accordingly, two graphs are edge-neighbors, if in one of them one edge is included/excluded independent of its label.

Definition 2 (Edge-neighboring Graphs) *Graphs* $G = V, E$ *and* $G^{'} = V^{'}, E^{'}$ *are* edge-neighbors *if* $V = V^{'}$ *and* $E^{'} = E - \{e\}$ *for some edge* $e \in E^{'}$.

Second, in accordance with the 'node privacy' definition of unlabeled graphs, two graphs are node-neighbors if one of them is obtained from the other by adding/removing one arbitrary node and all of its labeled edges.

Definition 3 (Node-neighboring Graphs) *Graphs* $G = V, E$ *and* $G^{'} = V^{'}, E^{'}$ *are* node-neighbors *if* $V^{'} = V - x$ *and* $E^{'} = E - \{v_1, l, v_2 \mid v_1 = x \vee v_2 = x\}$ *for some* $x \in V$.

In the third adaptation, the differential privacy guarantee is built on the notion of graphs that differ by a set of labeled outedges of a node. The intuition is, in some of the applications of edge-labeled directed graphs such as RDF, an entity is represented by its associations and particular associations, indicated by certain labels, 'uniquely' identify that entity.

Definition 4 (QL-Outedge Neighboring Graphs) *Let* QL *be a subset of* L. *Graphs* $G = V, E$ *and* $G^{'} = V^{'}, E^{'}$ *are* QL-outedge-neighbors *if* $V = V^{'}$ *and* $E^{'} = E - \{v_1, l, v_2 \mid v_1 = x \text{ and } l \in QL\}$ *for some* $x \in V$.

Example 1. Let $QL = \{b\}$, given the QL, in Fig 1, 2 graphs $G = V, E$ and $G^{'} = V^{'}, E^{'}$ are QL-outedge neighbors, because in $G^{'}$ for the vertex 'y', there does not exists any of the outedges with the labels in QL. Similarly in Fig 3, 4, let $QL^{'} = \{a, b\}$, in $G^{'''} = V^{'}, E^{'}$ for the vertex 'y' all its outedges with all the labels in $QL^{'}$ are excluded. So, graphs $G^{''}$ and $G^{'''}$ are $QL^{'}$-outedge neighbors.

Given the different definitions of what it means for two edge-labeled directed graphs to be neighbors, the privacy guarantee of a privatized mechanism is formalized as:

Definition 5 (Differential privacy for edge-labeled directed graphs) *A privatized mechanism* K *for edge-labeled directed graphs satisfies* ϵ-edge differential privacy *(respectively,* ϵ-node differential privacy*, or* ϵ-QL-edge-labeled differential privacy *for some* $QL \subseteq L$*), if for every pair of graphs* G *and* $G^{'}$ *that are edge-neighbors (respectively, node-neighbors, or* QL*-outedge neighbors), and for all* $S \subseteq RangeK$*, it holds that:* $PrKG \in S \leq e^{\epsilon} \times PrKG^{'} \in S$.

where the probabilities represent the random choices made by K *and* $RangeK$ *denotes the set of all possible outputs of* K.

4 Discussion and Future Outlook

Differential privacy for graph data offers a strong mathematical privacy guarantee for releasing graph statistics, but the semantics of the privacy protection rests on the definition of neighboring graphs. Thus, the choice of the definition of neighboring graphs impacts the privacy/utility trade-off offered by the privatized mechanism.

In the case of edge-neighbors (i.e Definition 2), edge disclosure is protected by the privatized mechanism. For some applications such as analysis on email communication graphs where

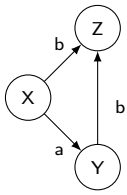


Fig. 1: G

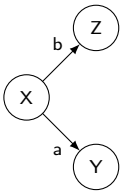


Fig. 2: $G^{'}$

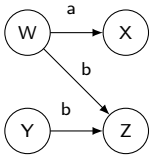


Fig. 3: $G^{''}$

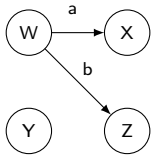


Fig. 4: $G^{'''}$

the relationships are sensitive, the edge-neighbors definition is sufficient. The node-neighbors definition (i.e Definition 3), mirrors the notion of neighboring databases formalized in Definition 1. However, the structural properties of graphs may introduce a huge gap between two neighboring graphs, consequently the function sensitivity is large, which need to be obfuscated by the privatized mechanism in order to protect the input graph that produces the result. The type of graph data analyses that a privatized mechanism can support under this definition yet producing accurate results may thus be limited. In many applications of edge-labeled directed graphs, the labels play a significant part in defining the relationships among the nodes. We assume that edges with labels from a particular domain-specific subset of all labels, 'uniquely' identify a node in an edge-labeled directed graph. Further, we focus on outedges of a node because it represents the contributions that this node makes to the graph dataset. Hence, this semantically captures the notion of an individual being in one graph but not in another graph similar to the private data represented as tuple in the relational databases. We propose that this definition of neighboring graphs offers another level of privacy protection than the edge-neighbors definition. Further, we hypothesize that under this definition the noise required to bridge the gap between the two neighboring graphs will be less than the node-neighbor definition, thus increasing the utility of the results returned by the privatized mechanism. Nevertheless, it would be interesting to study the type of graph statistics that are accurate and how accurate the results are under this scheme of things.

To test the hypothesis, as a next step we begin to focus on degree distribution as a particular graph statistics over edge-labeled directed graphs. Degree distribution of a graph gives a simplistic understanding of the structure of a graph. Degree distribution is a vector of real numbers that represent the degrees of the nodes in a graph. We plan to employ the privatized mechanism introduced by Dwork et al. [Dw06] to answer the degree distribution queries over edge-labeled directed graphs. Most importantly, we plan to study the privacy/utility trade-off of this privatized mechanism when the different neighbor definitions are chosen (i.e., Definition 2 versus Definition 3 versus Definition 4 with different QL). To this end, we plan to generate different edge-labeled graphs by systematically varying the structural characteristics, which constitute the datasets that are protected by the envisioned privatized mechanism. Based on these graphs, we aim to analyze the accuracy of the degree distribution query under the edge-neighbors versus the QL-outedge-neighbors (for different QL). Further, we aim to analyze the privacy/utility trade-off of our mechanism versus the Hay et al.'s

mechanism [Ha09], which supports degree distribution queries but requires a post processing step for improving the utility of the results. We also plan to evaluate the privacy/utility trade-off of our mechanism over a set of real-world graphs.

As a long-term goal, we move on to study other types of graph analyses, in particular, analyses that take into account the edge labels (e.g., in the case of degree distribution, to estimate the degree distribution that represent the edges with certain labels). From the results of these experiments that analyze the privacy/utility trade-off when different neighboring edge-labeled graph definitions are chosen, we aim to investigate different ways to optimize the privacy/utility trade-off in particular for QL-outedge neighbor definition.

Acknowledgments. I thank my advisor Olaf Hartig for the discussions and feedback that enable this work. I also thank Simone Fischer-Hübner for her feedback.

References

[AS00] Agrawal, Rakesh; Srikant, Ramakrishnan: Privacy-preserving Data Mining. In: Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data. 2000.

[AW89] Adam, Nabil R.; Worthmann, John C.: Security-control Methods for Statistical Databases: A Comparative Study. ACM Comput. Surv., 21(4):515–556, 1989.

[Be80] Beck, Leland L.: A Security Machanism for Statistical Database. ACM ToDS, 5(3), 1980.

[Ch05] Chawla, Shuchi; Dwork, Cynthia; McSherry, Frank; Smith, Adam D; Wee, Hoeteck: Toward Privacy in Public Databases. In: TCC. volume 3378, 2005.

[Dw06] Dwork, Cynthia; Mcsherry, Frank; Nissim, Kobbi; Smith, Adam: Calibrating noise to sensitivity in private data analysis. In: Proc. of 3rd TCC. 2006.

[Dw08] Dwork, Cynthia: Differential Privacy: A Survey of Results. In: TAMC: Proc. 5th Int. Conf.˙ 2008.

[Ha09] Hay, M.; Li, C.; Miklau, G.; Jensen, D.: Accurate Estimation of the Degree Distribution of Private Networks. In: 9th IEEE Int. Conf. on DM. 2009.

[Ka13] Kasiviswanathan, S. P.; Nissim, K.; Raskhodnikova, S.; Smith, A.: Analyzing Graphs with Node Differential Privacy. In: Proc. of 10th TCC (2013). 2013.

[NRS07] Nissim, K.; Raskhodnikova, S.; Smith, A.: Smooth Sensitivity and Sampling in Private Data Analysis. In: 39th ACM Symp. on Theory of Computing. 2007.

[Sw02] Sweeney, L.: k-anonymity: A model for protecting privacy. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10:557–570, 2002.

[TC12] Task, C.; Clifton, C.: A Guide to Differential Privacy Theory in Social Network Analysis. In: Int. Conf. on Advances in SN Analysis and Mining. 2012.

[ZPL08] Zhou, B.; Pei, J.; Luk, W.: A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. SIGKDD Ex. Nl., 2008.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 279

Turning the Table Around: Monitoring App Behavior

Nurul Momen[1]

Abstract: Since Android apps receive whitecard access through permissions, users struggle to understand the actual magnitude of app access to their personal data. Due to unavailability of statistical or other tools that would provide an overview of data access or privilege use, users can hardly assess privacy risks or identify app misbehavior. This is a problem for data subjects. The presented PhD research project aims at creating a transparency-enhancing technology that helps users to assess the magnitude of data access of installed apps by monitoring the Android permission access control system. This article will present how apps exercise their permissions, based on a pilot study with an app monitoring tool. It then presents a prototypical implementation of a networked laboratory for crowdsourcing app behavior data. Finally, the article presents and discusses a model that will use the collected data to calculate and visualize risk signals based on individual risk preferences and measured app data access efforts.

Keywords: App Behavior, Privacy Preservation, Transparency.

1 Introduction

User data is an important key for today's customer driven economy. An elusive, complex and service-centric business model took over the usual consumer-product model. In this new model, most of the users remain in wonderland consuming 'free' services while turning themselves into a product. In order to support the revenue model, mobile apps are being used to profile users to a great extent. Service providers are supposed to utilize such user data to deliver more customer-centric packages, but the probable and rather obvious privacy risks are usually broomed under the carpet. Apps are demanding too many permissions [Mo17], leaving the actual access to data opaque to the users, thus creating a data protection problem. Our project focuses on designing and developing transparency-enhancing tools (TETs) [He08] to uncover some of the risks in order to help users by making more informed decisions to protect their privacy.

The app markets are thriving with popularity metrics and crowdsourced user opinions which are very unlikely to be based on security and privacy factors. The rapid growth of the app market has outpaced the development of adequate transparency and control over user information. In the KAUdroid project we make an effort to ease users' remorse by utilizing TETs. In this paper, we introduce an app-behavior-analysis-model which includes a prototype app, a client-server architecture to document app behavior, a database and a visualization tool.

doi:10.18420/sicherheit2018_25

2 Background

A significant amount of research effort has been invested on the access control model of Android. In this section, a brief overview of related efforts is illustrated along with our main research goals.

2.1 Related Work

Several empirical studies pointed out that users face difficulties to perceive appropriate consequences of granting permissions to apps; for instance [Ke12, Pe12, KCS13]. Absence of regulation enforcement and technical measures to ensure the principle of least privilege is also held responsible for leaving sensitive user information in a vulnerable state [Fe11, HBM17]. In fact, apps are asking for more information than ever before [Au12, We12]. In order to aid the user in preserving privacy, we focus on investigating *how often* permissions are being accessed by the installed apps.

2.2 Research Goals

Since the introduction of runtime permission architecture, user consent is required during first-time use of the corresponding permission[3]. Granted privileges remain unchanged unless the user explicitly revoke them. Additionally, no statistics or qualitative information is available which could support reassessment of initial decisions. We argue that it diminishes the effectiveness of runtime permissions to some extent. As an initial inquiry, this project has performed a pilot study [Mo17], which found that apps are accessing granted resources more frequently and the interface is unable to offer neither quantitative, nor qualititative usage information to the user. Later on, we demonstrated the risks concerning partial identity generation from questionable and inconsistent resource usage patterns by apps [FM17]. Now we aim to investigate in a larger scale. Hence, measuring infrastructure is built to generate larger dataset in order to commence controlled experiments. This project is intended to answer following research questions: (1) How can privilege-induced privacy risks be communicated effectively to the user? (2) How can tools and methods assist users to benefit from ensuring the principle of least privilege for apps?

3 Model

In this section, we describe the proposed model for data analysis, determination of app's privacy impact factor (*PIF*) and app's risk score (*RS*). The model will be tested within an experimental lab that is being built for quantitative data collection purpose. The model is elaborated as following:

$M = (\delta, \alpha, \rho, \tau)$where,
δ = a finite set of participating devices/nodes;
α = a finite set of apps installed on the device;
ρ = a finite set of permissions requested by apps;

[3] `https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html`

τ = logged instances (timestamp) for permission usage.

For a given device δ_1, app α_1 and a time frame $\Delta\tau = (\tau_t - \tau_0)$, the total usage of permission ρ_1, is calculated as: $R_{\rho_1} = COUNT(\tau_i)$. Later on model M is used to accumulate data based on m number of different scenarios, S = s_1, s_2, ... s_m , where the elements refer to boolean values.

s_1 - *active/passive usage:* s_1 depends on whether the user is interacting with the app or not. It could be unlikely for a user to keep using an app throughout the day and night, but the app possesses corresponding permission to accumulate data (e.g. ACCESS_FINE_LOCATION) without any time constraint. We observed idle-time usage of the location permissions in our primary experiment. Thus we plan to document resource access events based on user interaction. This could reveal the app behavior throughout the day, week and month.

s_2 - *surrounding environment:* s_2 depends on whether the device is located in a crowded place or not. We hypothesize that apps could be greedy to discover user's peers in a crowded area which has the potential to extract the social graph of users. In our earlier work [FM17], the likelihood of partial identity (e.g. social graphs) extraction was presented. In this controlled scenario, devices are intended to be taken to crowded places (e.g. a restaurant, library, concert, shopping mall, etc.) and to observe the app behavior with respect to isolated usage.

s_3 - *motion and network connectivity:* stationary usage and wifi-only-connectivity of the device were the limitations for the previous study. With a view to overcome these shortcomings, scenario s_3 includes device deployment through short and long journeys with connectivity through wifi and mobile telephony. Devices will be kept and used during a journey (by train or by car) and document app behavior comparing with their stationary permission usage.

Further scenarios (s_4, s_5 and so on) can be defined and determined from individual preferences. Based on permission usage and scenarios, a *PIF* is calculated, which opens the option for accommodating tolerance and individual prioritization factor (p_m):

$$PIF = p_1.s_1 + p_2.s_2 + ... + p_m.s_m \tag{1}$$

If there are m number of $PIFs$ associated with corresponding scenarios, for a given app α with permission ρ running on a device δ for a selected time frame $\Delta\tau = (\tau_t - \tau_0)$, a risk score ($RS_\alpha$) is defined as following:

$$RS_\alpha = \sum_{i=0}^{m} R_{\rho_i}.PIF_{\rho_i} \pm E = \sum_{i=0}^{m} R_{\rho_i} \left(\sum_{j=0}^{n} p_j s_j \right) \pm E \tag{2}$$

The error margin (E) is yet to be defined and tested through analysis. Currently, we undertake data collection to fill the database with sufficient test data for populating the model under the scenarios. The experimental setup for testing the model is elaborated in the next section.

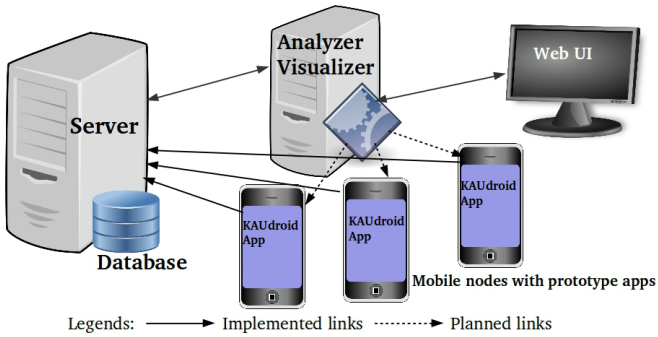


Fig. 1: Overall architecture.

4 System Architecture

In this section, we describe the system architecture of experimental setup that has been built to carry out controlled experiments [Ca18]. The system architecture is depicted in Fig. 1 and can be segmented into three different parts: a) logging, b) data collection, and c) analysis and visualization.

4.1 Logging

A prototype app has been developed that runs as a service on mobile devices. It is a monitoring app and is able to log each of the resource access events. The prototype is transparent to itself, and the log file also documents resource usage activities of the prototype app. The log is forwarded to the server through encrypted channel and stored remotely in the server.

AppOpsCommand is used to log the events[4]. The prototype checks for resource access events by each of the installed apps and writes respective events in a pre-defined format. The app stores the log records in a JSON file which contains three fields: 1) the name of the package/app, 2) the name of the accessed resource and, 3) the time of the resource access event. The following example shows a sample log record.

```
#root: adb shell appops get com.google.android.youtube
{"Package":"com.google.android.youtube","Permission":"READ_EXTERNAL_STORAGE",
"Timestamp":"Fri Mar 03 09:56:35 GMT+01:00 2017"}
```

4.2 Data Collection

The server is responsible for collecting, processing and storing logs sent from the mobile devices. The server is able to register participating devices/profiles. All the logs are first stored locally on the mobile device in a file (JSON) and is later sent to the server periodically (typically once a day). The server is also responsible for parsing the data and

[4] https://android.googlesource.com/platform/frameworks/base/+/android-6.0.1_r25/cmds/appops/src/com/android/commands/appops/AppOpsCommand.java;Accessed:2017-11-21

perform insert operations for the database. The database is consisted of three tables: 1) Main_info, 2) Time and 3) Coordinate. Figure 2 depicts the database schema. Package (or, app) and permission names are stored according to app's manifest. Time is stored in 'date+GMT+aa:aa:aa' format which enables it to handle data generated from any geographic location.

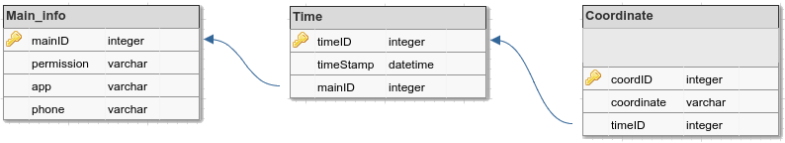


Fig. 2: Database Schema.

4.3 Analysis and Visualization

This part of the project is still in infant stage. Currently, a web-based interface can provide a brief summary to the resource usage events. We are also discussing the potentialities and properties of an effective visualization interface. For example, mean value of minimal resource usage could be calculated from the accumulated data within a given scenario and use it as a reference point to assess other permissions and apps. For similar scenarios ($S = S'$) and negligible discrepancy between intervals ($\Delta\tau \simeq \Delta\tau'$), condition for privacy-preserving behavior of an app is: $\overline{RS_\alpha} \geq RS'_\alpha$; where $\overline{RS_\alpha}$ = average risk score calculated from reference database and RS'_α = risk score of the target app.

Apart from the web interface, a personalized visualization feature is also included within the prototype app. In [MP17], we proposed a model to incorporate individual preference and to provide feedback based on self-defined threshold. The model includes a privacy measuring scale that possesses scalability within itself and it has the potential to offer additional *PIF* values. As the model is able to accommodate user-defined privacy preferences, it can be utilized to produce nudges in order to push the user toward privacy-preserving behavior. However, a meaningful threshold mechanism is yet to be defined and implemented. In Fig. 1, this is illustrated with dotted lines.

5 Discussion and Conclusion

From the pilot study, results and outcomes could be observed from two different angles. First, excessive permission usage and infraction from the principle of least privilege were highlighted [Mo17]. Second, a model for deriving partial identities from app permissions was presented which was induced from Pfitzmann and Hansen's terminology for privacy [PH10]. It also highlighted on the risks and likelihood of partial identity extraction [FM17].

Mobile app users pay with their personal data which is rather considered as an open secret. However, there is a hidden condition: price is undefined and uncontrolled; which leads to privacy risks. The installed apps possess unlimited access to sensitive user data without any time, frequency or volume constraint. This project aims to unearth such privilege induced risks and to provide a usable interface for assessing them. In order to inspect app behavior, the project collects log of resource access events, analyzes the data, displays summarized

statistics and seeks for a meaningful recommendation. It will allow the user to compare and be aware of privacy invasive behavior of apps and take initiatives to protect their private data.

References

[Au12] Au, Kathy Wain Yee; Zhou, Yi Fan; Huang, Zhen; Lie, David: Pscout: analyzing the android permission specification. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, pp. 217–228, 2012.

[Ca18] Carlsson, Adrian; Pedersen, Christian; Persson, Fredrik; Söderlund, Gustaf: KAUDroid : A tool that will spy on applications and how they spy on their users. Technical report, Karlstad University, 2018. Working paper, Januari 2018.

[Fe11] Felt, Adrienne Porter; Chin, Erika; Hanna, Steve; Song, Dawn; Wagner, David: Android permissions demystified. In: Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp. 627–638, 2011.

[FM17] Fritsch, Lothar; Momen, Nurul: Derived Partial Identities Generated from App Permissions. Open Identity Summit 2017, pp. 117–129, 2017.

[HBM17] Hammad, Mahmoud; Bagheri, Hamid; Malek, Sam: Determination and Enforcement of Least-Privilege Architecture in Android. In: Software Architecture (ICSA), 2017 IEEE International Conference on. IEEE, pp. 59–68, 2017.

[He08] Hedbom, Hans: A Survey on Transparency Tools for Enhancing Privacy. In: FIDIS. Springer, pp. 67–82, 2008.

[KCS13] Kelley, Patrick Gage; Cranor, Lorrie Faith; Sadeh, Norman: Privacy as part of the app decision-making process. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 3393–3402, 2013.

[Ke12] Kelley, Patrick Gage; Consolvo, Sunny; Cranor, Lorrie Faith; Jung, Jaeyeon; Sadeh, Norman; Wetherall, David: A conundrum of permissions: installing applications on an android smartphone. In: International Conference on Financial Cryptography and Data Security - FC2012 Workshops. Springer Berlin Heidelberg, pp. 68–79, 2012.

[Mo17] Momen, Nurul; Pulls, Tobias; Fritsch, Lothar; Lindskog, Stefan: How much Privilege does an App Need? Investigating Resource Usage of Android Apps. The Fifteenth International Conference on Privacy, Security and Trust (PST), 2017.

[MP17] Momen, Nurul; Piekarska, Marta: Towards Improving Privacy Awareness Regarding Apps' Permissions. Proceedings of the Eleventh International Conference on Digital Society and eGovernments - ICDS 2017, 2017.

[Pe12] Peng, Hao; Gates, Chris; Sarma, Bhaskar; Li, Ninghui; Qi, Yuan; Potharaju, Rahul; Nita-Rotaru, Cristina; Molloy, Ian: Using probabilistic generative models for ranking risks of android apps. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, pp. 241–252, 2012.

[PH10] Pfitzmann, Andreas; Hansen, Marit: Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. In: Designing privacy enhancing technologies. TU Dresden, pp. 1–9, 10-Aug-2010.

[We12] Wei, Xuetao; Gomez, Lorenzo; Neamtiu, Iulian; Faloutsos, Michalis: Permission evolution in the android ecosystem. In: Proceedings of the 28th Annual Computer Security Applications Conference. ACM, pp. 31–40, 2012.

H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 285

Usability von Security-APIs für massiv-skalierbare vernetzte Service-orientierte Systeme

Peter Leo Gorski[1]

Abstract: Kontemporäre Service-orientierte Systeme sind hochgradig vernetzt und haben zudem die Eigenschaft massiv-skalierbar zu sein. Diese Charakteristiken stellen im besonderen Maße Anforderungen an die Datensicherheit der Anwender solcher Systeme und damit primär an alle Stakeholder der Softwareentwicklung, die in der Verantwortung sind, passgenaue Sicherheitsmechanismen effektiv in die Softwareprodukte zu bringen.

Die Effektivität von Sicherheitsarchitekturen in service-orientierten Systemen hängt maßgeblich von der richtigen Nutzung und Integration von Security-APIs durch eine heterogene Gruppe von Softwareentwicklern ab, bei der nicht per se ein fundiertes Hintergrundwissen über komplexe digitale Sicherheitsmechanismen vorausgesetzt werden kann. Die Diskrepanz zwischen komplexen und in der Anwendung fehleranfälligen APIs und einem fehlenden Verständnis für die zugrundeliegenden Sicherheitskonzepte auf Seiten der Nutzer begünstigt in der Praxis unsichere Softwaresysteme. Aus diesem Grund ist die Gebrauchstauglichkeit von Security-APIs besonders relevant, damit Programmierer den benötigten Funktionsumfang effektiv, effizient und zufriedenstellend verwenden können.

Abgeleitet von dieser Problemstellung, konzentriert sich das Dissertationsvorhaben auf die gebrauchstaugliche Ausgestaltung von Security-APIs und den Herausforderungen die sich aus den Methoden zur Evaluation der Usability in typischen Umgebungen der Softwareentwicklung ergeben.

Keywords: Security-APIs; Usability; Usable Security; Softwareentwicklung

1 Einleitung

Application Programming Interfaces (dt. Schnittstellen zur Anwendungsprogrammierung) oder kurz *APIs*, sind aufgrund der fortschreitenden digitalen Transformation in allen Lebensbereichen und des hohen Vernetzungsgrades heutiger Technik allgegenwärtig. Der Grund dafür liegt in dem Konzept, das mithilfe von APIs umgesetzt werden kann: die einfache Nutzung bzw. *Wiederverwendung* von bereits verfügbaren *Funktionalitäten* durch *Abstraktion*. Bei Funktionalitäten handelt es sich z.B. um bereits entwickelte Programmteile, Implementierungen von mathematischen Lösungsverfahren, GUI-Elementen, Sicherheitsmechanismen, Hardwarefunktionen oder komplexe Daten wie aktuelle Wettervorhersagen.

[1] Technische Hochschule Köln, Data & Application Security Group, Betzdorfer Straße 2, 50679 Köln / Technische Universität Berlin, Quality and Usability Lab, Ernst-Reuter-Platz 7, 10587 Berlin peter.gorski@th-koeln.de

doi:10.18420/sicherheit2018_26

Die Menge an Funktionalitäten umfasst potenziell alle Problemstellungen, die durch Software gelöst werden können. Durch die Abstraktion in Form einer API reduziert sich die Komplexität, welche der Problemlösung bzw. Implementierung anhaftet. Dabei werden Informationen reduziert, die für die Nutzung der eigentlichen Funktionalität irrelevant sind, wodurch „einfach“ wiederverwendbare Softwarekomponenten entstehen. Diese können als Teillösungen weiterer Problemstellungen dienen. Damit ist es möglich, ein System aufzubauen, das sich aus aufeinander aufbauenden Softwarebausteinen zusammensetzt und mit jedem hinzukommenden Bauteil an Funktionalität und Komplexität zunimmt. APIs bilden die elementaren Bindeglieder dieses modularen Aufbaus.

Diesem Prinzip folgend, basieren wiederverwendbare Komponenten von Sicherheitssoftware auf komplexen digitalen Sicherheitskonzepten. Security-APIs werden vor diesem Hintergrund von spezialisierten Entwicklern entworfen und implementiert. Eine Security-API kann als Programmierschnittstelle definiert werden, die dem Softwareentwickler eine Sicherheitsfunktionalität zur Verfügung stellt, welche eine oder mehrere Sicherheitsregeln bei der Interaktion zwischen mindestens zwei Entitäten durchsetzt [GLI16]. Wenn es darum geht Sicherheitsmechanismen in diverse Softwareprodukte zu integrieren, greifen Programmierer auf diese Security-APIs zurück, insbesondere auch solche Entwickler, die nicht über ein fundiertes Hintergrundwissen über die eingesetzten Sicherheitsfunktionen verfügen. Die Sicherheit heutiger Software hängt daher vor allem von der effektiven Nutzung dieser Security-APIs durch eine heterogene Gruppe von Softwareentwicklern ab [Ge12, Fa13, Fa12, Na16, Ac17, LIG17].

2 Problemstellung

Eine wichtige Erkenntnis aus dem Forschungsgebiet *Usable Security und Privacy* ist, dass nutzerzentrierte Ansätze einen wichtigen Faktor für die kontinuierliche Verbreitung, Akzeptanz, Weiterentwicklung und besonders für die effektive, effiziente und zufriedenstellende Anwendung von Sicherheitskomponenten bilden [FHLIM10]. Werden Security-APIs nicht entsprechend den Anforderungen von Softwareentwicklern ausgestaltet, kommt es zur unwillentlichen oder auch willentlichen Fehl- bzw. Nichtbenutzung, wodurch die Sicherheitsdienste nicht mehr oder nur sehr eingeschränkt erbracht werden.

Softwareentwickler nehmen dabei eine Schlüsselrolle ein, denn der Fehler eines einzelnen Programmierers bei der Implementierung einer Sicherheitsfunktion hat durch die starke Vernetzung heutiger Systeme einen unmittelbar negativen Lawinen-Effekt auf die Sicherheit von Endanwendern. Daraus resultierende Sicherheitsprobleme sind z.B. im Bezug auf das Transport-Layer-Security-Protokoll (TLS) [DR08] aufgedeckt worden. Dies konnte für Programmierer ohne Sicherheitsexpertise [Na16, Ac17] aber auch für Sicherheitsexperten [Du17] belegt werden.

In diesen Fällen ist die Zertifikatsvalidierung von Softwareentwicklern fehlerhaft implementiert worden. Dadurch führt die Prüfung der Authentizität von empfangenen Daten zu

falschen Ergebnissen. Authentische und nicht authentische Quellen können somit nicht mehr unterschieden werden. Das Fehlen dieses grundlegenden Sicherheitsdienstes kann von einem Angreifer durch einen sogenannten Man-in-the-Middle Angriff [OW15] ausgenutzt werden, um die Vertraulichkeit und Integrität der ausgetauschten Daten aufzuheben. Betroffen waren bzw. sind Android und iOS Applikationen [Fa12, Fa13] und eine Vielzahl von SSL (Secure Socket Layer) – aus dem später TLS hervorgegangen ist – Libraries und Frameworks [Ge12].

Massiv-skalierbare vernetzte Service-orientierten Systemen adressieren eine hohe Anzahl von Endgeräten, welche von Endanwendern in diversen Bereichen des alltäglichen Lebens eingesetzt und genutzt werden. Bei einem Angriff auf diese Art von Systemen ist die Anzahl potenzieller Angriffsziele dementsprechend hoch. Als Beispiel findet der Architekturstil REST (Representational State Transfer) [Fi00], als Entwurfskonzept für solche Systemen, ubiquitär Anwendung in Domänen wie z.B. dem klassischen Internet, dem Internet der Dinge, der Industrie 4.0 oder der Gesellschaft 5.0 [CE17]. Diese Begriffe charakterisieren Vernetzung einhergehend mit einem stetigen Anstieg der Digitalisierung. Die daraus resultierenden Sicherheitsanforderungen müssen von Softwareentwicklern in die diversen Systeme implementiert werden. Eine übergeordnete Forschungsfrage der Usable Security Disziplin lautet daher: Wie lässt sich das Risiko ineffektiver Sicherheitsfunktionen durch geeignete Unterstützung der Softwareentwickler minimieren?

3 Stand der Forschung

Einige grundlegende Ansätze zur Beantwortung dieser Frage hat die Forschung auf dem Gebiet der allgemeinen API-Usability vorgeschlagen [Bl06, St09, Cl10, SK15]. Dabei wurden vor allem drei wichtige Faktoren identifiziert und adressiert: Das API-Design, Dokumentationen und Werkzeuge für den Umgang mit APIs. Ein vorgeschlagenes Modell der API-Usability gibt einen strukturierten Überblick über die bisherigen Erkenntnisse [GLI16].

Diese reichen jedoch nicht alleine aus, um die Gebrauchstauglichkeit von Security-APIs beschreiben zu können, denn aus dem Sicherheitskontext der *Security-APIs* ergeben sich zusätzlich besonders zu berücksichtigende Usability Aspekte [GS16, GLI16], wie z.B. der *Endanwenderschutz*, die *Prinzipientreue*, die *Einschränkbarkeit* oder die *Konfigurierbarkeit*, deren gebrauchstaugliche Ausgestaltung erst noch erforscht werden muss.

Dies gilt auch für spezifische Anforderungen, die sich für die *Uniform Interface* Richtlinie des Architekturstils REST ergeben, um z.B. in Benutzerstudien Berücksichtigung zu finden. Aus den genannten Gründen gilt dies insbesondere für die Gebrauchstauglichkeit von Security-APIs als Teil eines noch zu entwickelnden REST-Security Framework [LING17], deren Funktionalität die Sicherheit von REST-basierten Softwaresystemen gewährleisten soll. Ebenso sind Fragestellungen im Bezug auf die gebrauchstaugliche Ausgestaltung von Sicherheitsmechanismen in Entwicklungsumgebungen sowie von Dokumentationen offen. Aktuelle Arbeiten zeigen hier insbesondere Schwächen bei der Deckung des speziellen

Informationsbedarfs von Softwareentwicklern, bei der Implementierung von Sicherheitsfunktionalitäten auf [Ac16, LIG17].

Um damit verbundene Probleme zu adressieren, müssen Erwartungshaltungen und Informationsbedarfe von Softwareentwicklern erhoben und effektive Ansätze erforscht werden, wie Informationsflüsse zwischen den Entwicklern von Security-APIs und den Anwendern mittels Verknüpfungen von API-intrinsischen Informationen, der Informationsaufbereitung in Dokumentationen und Werkzeugen innerhalb von Entwicklungsumgebungen ermöglicht werden können. Dazu muss ebenfalls analysiert werden, ob es eine typische Entwicklungsumgebung für Softwareentwickler gibt, und wenn ja, wie diese aussieht oder ob eine Gruppierung verschiedener Ausprägungen möglich ist, um diese in Designprozessen und Methoden zur Evaluation berücksichtigen zu können. Bisher vorgeschlagene Evaluationsmethoden der allgemeinen Gebrauchstauglichkeit von APIs wurden von dem Forschungsgebiet der HCI (Human Computer Interaction) adaptiert [St09, Cl10, PFM10, SK15]. Spezielle Anforderungen an diese Methoden im Kontext der Usability Evaluation mit Softwareentwicklern müssen noch erforscht werden.

4 Forschungsfragen

Das Dissertationsvorhaben untersucht die aufgezeigten Fragestellungen bezüglich der gebrauchstauglichen Ausgestaltung von ubiquitär eingesetzten REST-Security Programmierschnittstellen zur effektiven und effizienten Verwendung durch Softwareentwickler. Dazu werden Aspekte wie das API Design, die Dokumentation und Werkzeuge zur Softwareentwicklung im Hinblick auf deren Einfluss auf die Gebrauchstauglichkeit untersucht. Als Grundlage dienen Erkenntnisse und Methoden aus den Bereichen der Usability-Forschung in Bezug auf generelle Programmierinteraktionen, die im Kontext von Sicherheitsbibliotheken transportiert bzw. angepasst, erweitert und evaluiert werden.

Das geplante Dissertationsvorhaben widmet sich zusammenfassend den folgenden Fragestellungen:

- Wie müssen Security-APIs ausgestaltet werden, damit diese von unerfahrenen bzw. nicht-spezialisierten Softwareentwicklern effektiv, effizient und zufriedenstellend verwendet werden können?
- Durch welche Methoden kann die Gebrauchstauglichkeit von Security-APIs evaluiert werden?
- Inwiefern können Unterstützungswerkzeuge zur Verwendung von Security-APIs (integriert in gängige Entwicklungsumgebungen) die Gebrauchstauglichkeit beeinflussen?
- Was für einer Taxonomie folgt das Forschungsfeld der Gebrauchstauglichkeit von Security-APIs?

Danksagung

Das Projekt ULS3 (Ultra-Large Scale Systems Security) wird unter dem Förderkennzeichen 13FH016IX6 im Förderprogramm “Forschung an Fachhochschulen” vom Bundesministerium für Bildung und Forschung (BMBF) gefördert.

Literaturverzeichnis

[Ac16] Acar, Yasemin; Backes, Michael; Fahl, Sascha; Kim, Doowon; Mazurek, Michelle L.; Stransky, Christian: You Get Where You’re Looking for: The Impact of Information Sources on Code Security. In: IEEE Symposium on Security and Privacy. S&P ’16, 2016. doi: 10.1109/SP.2016.25.

[Ac17] Acar, Yasemin; Backes, Michael; Fahl, Sascha; Garfinkel, Simson; Kim, Doowon; Mazurek, Michelle; Stransky, Christian: Comparing the Usability of Cryptographic APIs. In: IEEE Symposium on Security and Privacy. S&P ’17, 2017. http://www.ieee-security.org/TC/SP2017/papers/161.pdf.

[Bl06] Bloch, Joshua: How to design a good API and why it matters. In: OOPSLA ’06, Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications. 2006. doi: 10.1145/1176617.1176622.

[CE17] CEBIT: Society 5.0: Japan treibt auf der CeBIT 2017 die Digitalisierung voran. 2017. Online: https://www.cebit.de/de/news-trends/news/society-5-0-japan-treibt-auf-der-cebit-2017-die-digitalisierung-voran-722 (13.02.2018).

[Cl10] Clarke, Steven: How Usable Are Your APIs? In (Oram, Andy; Wilson, Greg, Hrsg.): Making software: What really works, and why we believe it, Theory in practice, S. 545 – 565. O’Reilly, Beijing, 1. Auflage, 2010.

[DR08] Dierks, T.; Rescorla, E.: RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2. Proposed Standard,, Internet Engineering Task Force (IETF), 2008.

[Du17] Durumeric, Zakir; Ma, Zane; Springall, Drew; Barnes, Richard; Sullivan, Nick; Bursztein, Elie; Bailey, Michael; Halderman, J. Alex; Paxson, Vern: The Security Impact of HTTPS Interception. In: The Network and Distributed System Security Symposium 2017. NDSS ’17, 2017. doi: 10.14722/ndss.2017.23456.

[Fa12] Fahl, Sascha; Harbach, Marian; Muders, Thomas; Smith, Matthew; Baumgärtner, Lars; Freisleben, Bernd: Why Eve and Mallory Love Android: An Analysis of Android SSL (In)Security. In: 19th ACM Conference on Computer and Communications Security. CCS ’12, 2012. doi: 10.1145/2382196.2382205.

[Fa13] Fahl, Sascha; Harbach, Marian; Perl, Henning; Koetter, Markus; Smith, Matthew: Rethinking SSL Development in an Appified World. In: 20th ACM SIGSAC Conference on Computer and Communications Security. CCS 2013, 2013. doi: 10.1145/2508859.2516655.

[FHLIM10] Fischer-Hübner, Simone; Lo Iacono, Luigi; Möller, Sebastian: Usable Security und Privacy. Datenschutz und Datensicherheit - DuD, 34(11):773–782, 2010. doi: 10.1007/s11623-010-0210-4.

[Fi00] Fielding, Roy Thomas: Architectural Styles and the Design of Network-based Software Architectures. Dissertation, University of California, 2000.

[Ge12] Georgiev, Martin; Iyengar, Subodh; Jana, Suman; Anubhai, Rishita; Boneh, Dan; Shmatikov, Vitaly: The Most Dangerous Code in the World: Validating SSL Certificates in Non-browser Software. In: 19th ACM Conference on Computer and Communications Security. CCS '12, 2012. doi: 10.1145/2382196.2382204.

[GLI16] Gorski, Peter Leo; Lo Iacono, Luigi: Towards the Usability Evaluation of Security APIs. In: Tenth International Symposium on Human Aspects of Information Security & Assurance. HAISA 2016, 2016. http://www.cscan.org/openaccess/?paperid=287.

[GS16] Green, Matthew; Smith, Matthew: Developers are Not the Enemy!: The Need for Usable Security APIs. IEEE Security & Privacy, 14(5):40–46, 2016. doi: 10.1109/MSP.2016.111.

[LIG17] Lo Iacono, Luigi; Gorski, Peter Leo: I Do and I Understand. Not Yet True for Security APIs. So Sad. In: The 2nd European Workshop on Usable Security. EuroUSEC '17, 2017. doi: 10.14722/eurousec.2017.23015.

[LING17] Lo Iacono, Luigi; Nguyen, Hoai Viet; Gorski, Peter Leo: On the Need for a General REST-Security Framework. ACM Transactions on the Web (TWEB), 2017. In review process.

[Na16] Nadi, Sarah; Krüger, Stefan; Mezini, Mira; Bodden, Eric: Jumping Through Hoops: Why Do Java Developers Struggle with Cryptography APIs? In: The 38th International Conference on Software Engineering. ICSE '16, 2016. doi: 10.1145/2884781.2884790.

[OW15] OWASP: Man-in-the-middle attack. 2015. Online: https://www.owasp.org/index.php/Man-in-the-middle_attack (13.02.2018).

[PFM10] Piccioni, Marco; Furia, Carlo A.; Meyer, Bertrand: An Empirical Study of API Usability. In: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM '13, 2013-10. doi: 10.1109/ESEM.2013.14.

[SK15] Scheller, Thomas; Kühn, Eva: Automated Measurement of API Usability: The API Concepts Framework. Information and Software Technology, 61:145–162, 2015. doi: 10.1016/j.infsof.2015.01.009.

[St09] Stylos, Jeffrey: Making APIs More Usable with Improved API Designs, Documentation and Tools. Dissertation, Carnegie Mellon University, 2009. http://www.cs.cmu.edu/ natprog/papers/CMU-CS-09-130-Stylos-Dissertation.pdf.

Autorenverzeichnis