RUHR
UNIVERSITÄT
BOCHUM

RUB

Analyzing Privacy and End User Information Exposure in Digital Communication Environments

DISSERTATION

zur Erlangung des Grades eines Doktor-Ingenieurs
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

vorgelegt von

Theodor Schnitzler
geboren in Essen

Bochum, Dezember 2022

Tag der mündlichen Prüfung: 24. Juni 2022

Gutachter:innen

Prof. Dr. Christina Pöpper, New York University Abu Dhabi, V.A.E.

Prof. Dr. Markus Dürmuth, Leibniz-Universität Hannover, Deutschland

Prof. Dr. Tim Güneysu, Ruhr-Universität Bochum, Deutschland

Abstract

By using ever-present digital applications that have turned into inevitable tools for various everyday purposes, users leave countless traces of personal data in such applications and on the Internet. In this work, we explore what kind of information is exposed in several eminently important use cases that apply to millions of users on a daily basis. This covers information that users actively share with other persons, for example by posting contents in online social networks, as well as information that originates from the use of specific applications such as mobile messengers, and that can be obtained by others, although not intended by the respective users.

Regarding actively shared information, the longitudinal management of information availability is a crucial aspect, specifically allowing controlled expiration of data under certain conditions. In this context, we provide an extensive systematization of existing research on longitudinal online data management. We contrast technical concepts for exposure reduction and insights from user studies, and identify and analyze research gaps between the two sides. Academic proposals for data expiration that provide guarantees for technical security have been well-researched but do not appropriately reflect users' needs for handling their data. Thus, we propose a fundamentally different approach, taking into account multiple perspectives including legal aspects, in which users and online services can formally agree on data lifetime ending with the help of smart contracts. Such agreements allow for more flexible specification of data handling, and can also be used to incentivize that online services realize data expiration.

In contrast to data that is actively shared, also the use of specific online applications has the potential to expose certain user information to others, despite not being intended. The anonymity network Tor allows concealing one's identity when using the Internet. However, analyzing

network traffic of users basically re-enables tracking their online activities, which works highly efficiently under controlled conditions. In this context, we examine how realistic such well-known deanonymization attacks are when we take into account the actual network and Internet infrastructure. Finally, we analyze an unexpected timing side channel in mobile messenger apps, which can be used to spy on one's contacts. The time it takes for sent messages to be confirmed differs between everyday locations due to characteristics of different Internet connections. Observing these distinguishable timings thus enables a client to determine the current whereabouts of a target user simply upon sending them messages.

Kurzfassung

Durch die Nutzung allgegenwärtiger digitaler Anwendungen, die heutzutage für verschiedene alltägliche Zwecke nahezu unabdingbar sind, hinterlassen Nutzer:innen unzählige Spuren persönlicher Daten in diesen Anwendungen und im Internet. In dieser Arbeit wird untersucht, welche Arten von Informationen in verschiedenen Anwendungsfällen, die Millionen von Nutzer:innen täglich betreffen, preisgegeben werden. Dies umfasst sowohl Informationen, die aktiv anderen Menschen zur Verfügung gestellt werden, z.B. durch das Teilen von Inhalten in sozialen Netzwerken, als auch Informationen, die allein durch die Nutzung bestimmter Anwendungen, z.B. mobile Messenger, offenbart und von anderen erlangt werden können, auch wenn es von den entsprechenden Nutzer:innen nicht beabsichtigt ist.

Hinsichtlich aktiv geteilter Informationen ist das Langzeitmanagement der Informationsverfügbarkeit ein wichtiger Anwendungsfall, im Besonderen das kontrollierte Vergessen von Daten unter bestimmten Bedingungen. In diesem Zusammenhang wird eine umfassende Systematisierung existierender Forschungsarbeiten zum Thema Langzeitmanagement von Online-Daten durchgeführt. Dabei werden verfügbare technische Konzep-te zur Reduzierung der Sichtbarkeit den Erkenntnissen aus Nutzerstudien zum Umgang mit Online-Daten gegenübergestellt und Forschungslücken zwischen beiden Seiten identifiziert und analysiert. Akademische Konzepte für den Verfall von Daten, die zwar technisch Sicherheit garantieren, spiegeln die Bedürfnisse von Nutzer:innen im Hinblick auf den Umgang mit ihren Daten aber nicht angemessen wider. Deshalb wird ein fundamental anderer Ansatz vorgeschlagen, der verschiedene Perspektiven, unter anderem juristische Aspekte, mit einbezieht und mit dem Nutzer:innen und Online-Dienste das Ende der Verfügbarkeit von Online-Daten mithilfe von Smart Contracts vertraglich festlegen können. Derartige Vereinbaren ermöglichen eine flexiblere Spezifizierung des Man-

agements der Daten, und können auch so eingesetzt werden, dass für den Online-Dienst ein Anreiz geschaffen wird, den Verfall der Daten zu realisieren.

Im Gegensatz zum aktiven Teilen von Daten erlaubt auch die reine Nutzung bestimmter Online-Anwendungen, dass Außenstehende Informationen über Nutzer:innen gewinnen können. Der Anonymisierungsdienst Tor erlaubt, die eigene Identität bei der Nutzung des Internets zu verschleiern. Durch die Analyse von Netzwerkverkehr können die Aktivitäten von Nutzer:innen prinzipiell jedoch wieder nachvollzogen werden, was unter kontrollierbaren Rahmenbedingungen sehr gut funktioniert. In diesem Kontext wird untersucht, wie realistisch derartige Deanonymisierungsangriffe unter Berücksichtigung der echten Netzwerkinfrastruktur sind. Weiterhin wird im Fall von mobilen Messengern analysiert, in welchem Ausmaß Nutzer:innen von ihren Kontakten über einen unerwarteten auf Zeiten basierenden Seitenkanalangriff überwacht werden können. Die Zeit, die vergeht, bis eine gesendete Nachricht auf dem Sendegerät als zugestellt bestätigt wird, unterscheidet sich zwischen verschiedenen alltäglichen Orten aufgrund der Eigenschaften unterschiedlicher Internetanbindungen. Das Beobachten solcher unterscheidbarer Zeiten ermöglicht daher, allein durch das Senden von Nachrichten, den aktuellen Aufenthaltsort eines Empfangsgeräts zu bestimmen.

Acknowledgements

First, I would like to express my gratitude to my co-advisors Prof. Dr. Christina Pöpper and Prof. Dr. Markus Dürmuth for all their guidance and support and at the same time granting me freedom to explore and pursue research topics on my own.

I am grateful to work with Prof. Dr. Katharina Kohls on numerous projects, for countless discussions on papers, and for a lot of helpful advice for growing as a researcher. In addition, I would also like to thank Prof. Dr. Thorsten Holz for sharing his thoughts in fruitful project discussions, and Prof. Dr. Tim Güneysu for evaluating my PhD thesis.

I would like to thank again Prof. Dr. Christina Pöpper for hosting my research visit at New York University Abu Dhabi, and Shujaat Mirza, Evangelos Bitsikas, and all other members of the Cyber Security and Privacy Lab for welcoming me in their group and turning my visit into an intense but joyful, unforgettable, and truely unique experience.

A vital part of my PhD life was the enthusiasm and company of my colleagues in the Information Security and Mobile Security groups. All sorts of shared activity, including breakfasts, a hell lot of cake, spekulatius smoothie, collective sports events, currywurst-all-you-can-eat, and cocktail nights created a lot of fun and memorable moments. In particular, I would like to thank Dr. Maximilian Golla for being an extraordinarily Satisfactory office mate taking care of that the gummibärchen jars never run empty, Dr. Kai Jansen for pushing me to physical limits and beyond without becoming a Human Fall Flat, Jan Wiele for being Heroes Of grinding coffee beans for our french press and Dr. David Rupprecht for securing our mobile networks facing The Storm of threats, Dr. Lea Schönherr for suggestions and tips for building the Definitive Edition of a garden Empire, Florian Farke for introducing me to the Borderlands of strangest TEX packages, Philipp Markert for a jump moment I will never be Getting Over It, and Franziska Herbert, Marvin Kowalewski,

and Leona Lassak for being Among Us to continue the journey of our group.

All the work we created and published over the past years would have never been possible without the incredible commitment of all additional colleagues, collaborators, and co-authors: Dr. Christine Utz, Dr. Martin Degeling, Dr. Steffen Becker, Vera Rimmer, Tom Van Goethem, Abel Rodríguez Romero, and all the others.

Finally, I would like to thank my family for all kinds of endless support over the past decades, and in particular my wife Lena for her love, and for staying with me on the same Raft for all the years en in de toekomst.

Contents

1
Introduction

Contents

1.1. Motivation

Over the past decades, the Internet has become an integral part of our daily lives, offering a wide range of tools used for private and professional purposes. The number of global Internet users was estimated 4.88 billions in October 2021 [96] (i. e., 62 % of the world's population) and, therefore, it has more than doubled over the past ten years [201]. The most fundamental forms of Internet use include communication and information sharing between individuals or groups of individuals. Platforms and services such as Facebook, YouTube, and WhatsApp each connect more than two billion monthly active users around the world [203].

In using online services and devices, people leave detailed traces of their online activities, both deliberately and unintended. Users *deliberately* share information with others, e. g., by communicating through a messaging application, or by posting contents made available for their peers on social network sites. In one second, there are almost 10,000 tweets posted on Twitter, and more than 1000 photos posted on Instagram [165], many of them disclosing particular details of private lives. Keeping track of all information shared with others, and controlling their dissemination throughout their entire time of availability is a challenging task. In this context, the first part of this thesis focuses on *managing self-published online data*, particularly addressing challenges regarding longitudinal aspects of data, and managing their lifetime and availability. For the general public, sharing personal contents online has yet become available with the appearance of online social networks over the past 15 to 20 years. Therefore, knowledge about longitudinal effects of ever-present personal data is still at a comparably early stage and limited, which makes it very important to shed light on the topic and to contribute knowledge to this field of research.

Besides information that is deliberately shared with others, there is also information that users disclose *unintended*, often without their knowledge, and that can be obtained by others observing their interaction

with a specific application. The analysis of meta information has received considerable public attention, e. g., when companies aggregated and evaluated Facebook user data for micro-targeting purposes in the context of elections in the United States [121]. In a different use case, German telecommunication providers have passed aggregated cellular location data of their customers to governmental institutions in mid-2020 to better monitor the spread of the coronavirus pandemic in its early phase [174]. While these are two prominent but rather generic examples for the evaluation of meta information, we examine how the analysis of network traffic enables *usage-driven information revelation* in the second part of this thesis. More precisely, we focus on the practicality of traffic analysis in real-world communication environments, and their consequences for end user privacy. In this part, we consider two types of applications in which unintended information revelation has different impacts – the anonymity network Tor represents technology users explicitly use for privacy purposes, whereas mobile messengers are widely adopted by large numbers of people for everyday communication.

Managing Self-Published Online Data When personal contents are shared in online environments, they can be copied, forwarded, and arbitrarily processed and transferred by other individuals. As soon as contents are made accessible to others, their owners are not able to control the contents' onward dissemination on the Internet and often do not actively keep track of them [40, 97, 129]. This becomes particularly challenging, when users aim to manage the availability of their contents in the long term, e. g., to reduce their visibility in retrospect, or to end the lifetime of data by eventually revoking access to it.

While there are experiences that are worth remembering and keeping shared with others for long periods of time, there is also a lot of content that was not meant to be made available permanently when it was initially shared. In addition to that, user preferences regarding the value of contents to be shared may change. Without appropriate measures for

dissemination control and longitudinal management, contents can persist very long periods of time without receiving notable attention, yet turn up again in inconvenient moments [69,172]. Unpopular contents remaining online prevents people from overcoming their past mistakes, and thwarts giving them a second chance [25, 59]. Since online contents can be easily accessed and retrieved from anywhere in the world at any time, the open nature of the Internet requires fundamentally different concepts and approaches for archiving and forgetting. Those concepts that were well-established for analogous records physically stored in a specific location do not hold for ever-accessible digital goods [119]. Research in this field has made several attempts to provide solutions for revocation of online data [8,35,65,152,166,250]. However, these proposals still have unrealistic limitations, are not fully elaborated, and not easy enough to use. It has been well known for years that comprehensiveness is a key factor for the adoption of a particular application [3, 101, 239]. In the first part of this thesis, we will address this topic from technical and user perspectives.

Usage-Driven Information Revelation Large parts of our daily lives involve the use of different types of online services, not only for communication and information exchange, but also for shopping online, entertainment purposes such as streaming movies, and even when making electronic payments in a physical store. However, interactions with online applications and characteristics of their related network transmissions reveal a lot of information about users to third parties. For example, Internet Service Providers or operators of DNS services can usually see which online services a user interacts with.

Anonymity systems such as Tor [212] allow users to protect their identities and their online activities from being directly monitored by external observers. However, even with such a privacy-preserving mechanism in place, network transmissions leak sensitive information about clients, such that their identities and the services they use can still be determined.

Such *traffic analysis attacks* [164] are well-studied and work quite well from a technical perspective [48, 82, 99, 132, 137, 138, 170]. Since evaluations of traffic analysis techniques require access to all network transmissions of interest to deliver accurate results, their operational requirements, i. e., actually having access to transmissions are often neglected. Therefore, additional evaluations are necessary in order to determine what information about users can actually be derived under realistic assumptions in practical scenarios. Whereas Tor is the most prominent target for traffic analysis, their techniques can also be applied to other environments such as instant messengers, and used to learn information about their users. In this thesis, we analyze the practicality of traffic analysis in two case studies, covering both Tor as a particularly privacy-preserving system, as well as mobile instant messengers representing widely-adopted everyday-use online applications.

1.2. Contributions

In this thesis, we contribute knowledge to research on privacy and data protection by analyzing end user information exposure in a variety of online applications. Part I addresses self-published online data, i. e., data that users willingly share with others in online applications. In this domain, we focus on longitudinal management and the lifetime ending, i. e., revocation of such data, particularly aiming to reduce the existing gap between academic approaches for data revocation and the way end users interact with it in real-world applications. Part II addresses information that can be derived about users when they use a particular application. We consider two types of applications – whereas *Tor* represents technologies explicitly utilized for privacy purposes such as identity protection, *instant messengers* are essential tools for and prevalent in everyday communication within a large and heterogeneous user base. Figure 1.1 provides an overview of the aspects covered in this work. In the following, we briefly introduce each topic covered in this thesis.

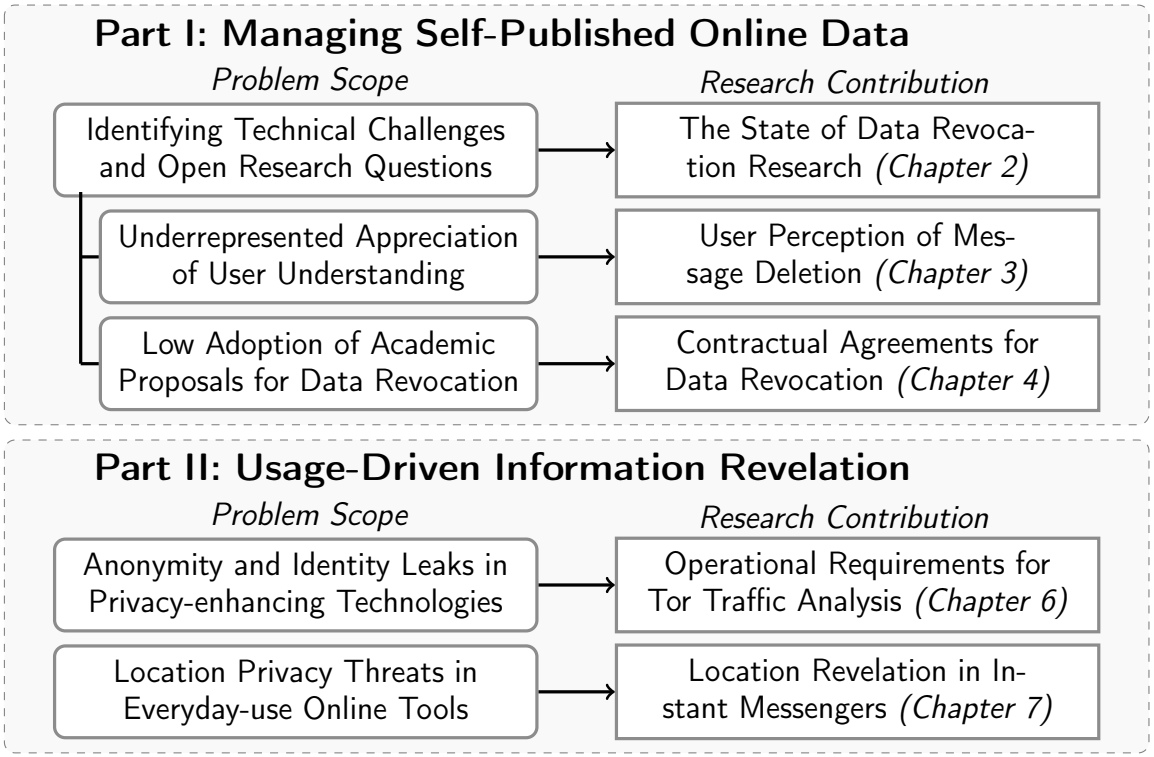

Figure 1.1. End user information exposure in digital communication environments.

1.2.1. Managing Self-Published Online Data

In the first part of this work, we focus on data that users send to or actively share with others through online applications and how the availability of such data can be managed longitudinally. Our contributions in this part include an extensive review of data revocation research from *user* and *technical* perspectives, along with identifying challenges that are not yet resolved. From the *user* perspective, we subsequently study how the proper design of interfaces can facilitate better user understanding of the effects of data deletion. In addition, we propose a fundamentally different *technical* approach for data revocation based on contractual agreements.

The State of Data Revocation Research There is a large body of research covering longitudinal management and revocation of self-published online data. However, such works either focus on the technical

feasibility of privacy management, or study how users interact with tools to manage the availability of their online contents. The first contribution of this thesis comprises a systematic review of previous research in this area over the past ten years, and provides taxonomies for both sides – the technical feasibility of longitudinal online privacy management and user interaction with it. First, we systematize how users interact with online privacy management tools by identifying different usage patterns, reasons for users to limit the visibility of their online contents, and their desires for facilitating interaction with online services to better fit their needs. Second, we also provide a systematization of research proposals for exposure reduction or revocation of personal content that is shared online, including the use cases they were designed for, the adversarial models they protect against and the protection mechanisms they use. Based on the analysis of previous works on both sides, we identify conflicts between them, namely incorrect, incomplete, or missing technical realizations of, e. g., user desires. Such conflicts include mismatches between the use cases a specific application and their privacy management mechanisms have been designed for, or how users actually interact with this specific application when managing the privacy of their contents. We derive open challenges and research questions to be addressed in future work. Our work is meant to facilitate the development of privacy-enhancing technologies that enable users to better manage the longitudinal privacy configuration of their online contents and that better fit their needs and desires.

The contributions of this work originate from a shared first author publication at PETS 2021 in collaboration with Shujaat Mirza, Markus Dürmuth, and Christina Pöpper. In particular, the author of this thesis contributed the systematization of user studies, Shujaat Mirza contributed the systematization of technical approaches, and the remaining sections were contributed equally.

User Perception of Message Deletion When WhatsApp, the mobile instant messengers with the largest user base world-wide, introduced a new feature to delete messages from all devices involved in a messaging conversation, the application let users explicitly select whether a message should be only deleted from the sender's device, or also from the receivers' devices. We study deletion features of 17 popular messengers, finding different functionalities, some of which do not clearly explain the effects of deletion.

Our results of a between-subjects user-study with 125 participants in WhatsApp, Facebook Messenger, and Skype contributes knowledge about how users interact with deletion functions in messenger apps, and their preferences for and attitudes towards message deletion on the receiver's side. We identify various reasons for users to delete messages, ranging from deleting and instantly re-sending messages due to editing or mis-spelling issues, to completely revoking messages that they regret or consider inappropriate in retrospect. We also find that 40 % of users prefer to be able to select if a message should be deleted on their own device only, or from all devices in a conversation, for each message individually. Finally, our study reveals that users can significantly better assess the effects of deletion, when it is clearly explained in the user interface, therefore, confirming previous findings.

The contributions of this work originate from a first author publication in the Journal of Cybersecurity in collaboration with Christine Utz, Florian M. Farke, Christina Pöpper, and Markus Dürmuth. In addition, Henry Hosseini and Eduard Leonhardt helped with collecting data for the user study.

Contractual Agreements for Data Revocation As one result of our systematic analysis of data revocation research shows, mechanisms for longitudinal privacy management are mainly time-based and do not fit user needs that are more complex and include multiple facets such as contents, audience, and contexts. Moreover, since many approaches to

data revocation involve cryptographic operations for access to contents, their practical deployment remains limited.

As the third contribution of this work, we propose a fundamentally different approach to data revocation. Instead of technically enforcing deletion after a certain amount of time has elapsed, our proposal builds on contractual agreements between users who upload a certain piece of content, and providers who make these contents available on their platforms. In such agreements, the involved parties can specify conditions for content expiration beyond time-based conditions, and penalties for a provider who does not follow the specification. Agreements can be established for newly published data, but also added retroactively for contents that have already been published. We provide an overview of the design space for contract-based data revocation and demonstrate the feasibility of our approach by implementing a prototype in the form of an Ethereum smart contract that can be publicly deployed.

The contributions of this work originate from a first author publication at IFIP SEC 2019 in collaboration with Markus Dürmuth and Christina Pöpper.

1.2.2. Usage-Driven Information Revelation

In the second part of this work, we focus on information about users that can be derived when they use specific online applications. However, such information is neither meant to be shared by users, nor intended to be revealed by the respective applications but can be derived by analyzing patterns of network traffic that is generated while the application is used. We showcase two scenarios for threats to user privacy in different types of applications, (i) in privacy-enhancing technologies that are explicitly used to ensure a high-level of privacy in online activities, i. e., Tor, and (ii) in tools widely adopted for everyday communication, i. e., mobile instant messengers.

Tor Exit Prediction The use of Tor helps users to reduce their own information exposure during their online activities. By separating client and server side traffic, users remain anonymous online, i. e., their identity or the information which online services they use can be protected from external parties. However, analyzing specific characteristics of network traffic, e. g., timings or packet sizes, at different points in the connection between client and server allows to de-anonymize connections through Tor. While state-of-the-art attacks work with high precision from a technical perspective, the operational requirements for these attacks such as access to traffic streams are often neglected, leaving their practicality in the actual network infrastructure a blind spot.

In this context, we contribute knowledge about the practicality of traffic analysis attacks in Tor under real-world requirements. To this end, we introduce three types of attacks that can be conducted as preliminary steps before attempting to run extensive traffic analyses. All the attacks we present and their underlying threat vectors are rooted in core mechanisms and defensive features in Tor, i. e., they are hard to mitigate. The attacks enable adversaries to determine whether a target client uses a specific connection under adversarial control, i. e., if the operational requirement for a subsequent traffic analysis is fulfilled. In addition, adversaries can tamper with Tor circuit establishment mechanisms to increase their chances of having access to client traffic. We simulate these attacks based on data derived from empirical measurements in the live Tor network to demonstrate their feasibility and to emphasize their consequences for follow-up traffic analysis attacks, enabling adversaries to uncover the identities of anonymous Tor clients under realistic requirements.

The contributions of this work originate from a first author publication at EuroS&P 2021 in collaboration with Christina Pöpper, Markus Dürmuth, and Katharina Kohls.

Location Revelation in Instant Messengers Mobile instant messaging enables users to communicate with others from anywhere in the world whenever their device is connected to the Internet. In order to inform users if messages they sent have reached their destination, messaging applications use status delivery icons alongside each message. The realization of this useful feature induces additional network transmissions between the devices and the messenger servers that can be identified using classical traffic analysis methods. These transmissions expose additional information about users involved in a communication with unexpected consequences for their location privacy.

As the last contribution of this thesis, we demonstrate how a timing side channel in message status notifications can be used to derive information about the locations of messenger users. By measuring the time between sending a message and retrieving the delivery notification from the receiver, the message sender can determine whether or not the receiver is at a specific location. In this context, we conduct a series of experiments that involves sending messages between devices across different countries in Europe and the Middle East and measuring and evaluating the timings of message delivery status notifications. As our evaluations show, a messenger user can, after a training phase, determine the correct receiver country out of the four countries in our experimental setup with 80 % accuracy after sending five WhatsApp messages. However, the timing side channel also persists on a considerably more fine-grained level. Different locations within the same city can also be distinguished quite well, in some cases with more than 90 % accuracy. We demonstrate that not only WhatsApp but also privacy-friendly messengers such as Signal and Threema are prone to this information leak.

The contributions of this work originate from a first author publication at NDSS 2023 in collaboration with Katharina Kohls, Evangelos Bitsikas, and Christina Pöpper. In addition, Marvin Kowalewski, Leona Lassak, Philipp Markert, Sarah Pardo, and Lena Schnitzler helped with data collection.

1.3. List of Publications

The main parts of this thesis are based on peer-reviewed publications. The research described in these publications is a result of collaborations with colleagues and research project members and includes work that was conducted during a research visit at New York University Abu Dhabi.

1) **T. Schnitzler**, S. Mirza, M. Dürmuth, C. Pöpper, "SoK: Managing Longitudinal Privacy of Publicly Shared Personal Online Data," in *Proceedings on Privacy Enhancing Technologies (PETS '21)*, vol. 2021, no. 1, pp. 229–249, Sciendo, Nov. 2020.

2) **T. Schnitzler**, M. Dürmuth, C. Pöpper, "Towards Contractual Agreements for Revocation of Online Data," in *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 Int'l Conference (SEC '19),* pp. 374–387, Springer, Jun. 2019.

3) **T. Schnitzler**, C. Utz, F. M. Farke, C. Pöpper, and M. Dürmuth, "Exploring User Perceptions of Deletion in Mobile Instant Messaging Applications," *Journal of Cybersecurity*, vol. 6, no. 1, pp. 1–15, Oxford University Press, Jan. 2020.

4) **T. Schnitzler**, C. Pöpper, M. Dürmuth, K. Kohls, "We Built This Circuit: Exploring Threat Vectors in Circuit Establishment in Tor," in *IEEE European Symposium on Security and Privacy (EuroS&P '21)*, pp. 319–336, IEEE, Sept. 2021.

5) **T. Schnitzler**, K. Kohls, E. Bitsikas, C. Pöpper, "Hope of Delivery: Extracting User Locations From Mobile Instant Messengers," in *The Network and Distributed System Security Symposium (NDSS '23)*, Feb. 2023 (*to appear*).

During the time of this dissertation, the author also contributed to the following works, including peer-reviewed publications and unpublished material that is currently under review.

6) **T. Schnitzler**, C. Utz, F. M. Farke, C. Pöpper, and M. Dürmuth, "User Perception and Expectations on Deleting Instant Messages – or – "What Happens If I Press This Button?"," in *European Workshop on Usable Security (EuroUSEC '18)*, pp. 1–9, The Internet Society, Apr. 2018.

7) M. Golla, **T. Schnitzler**, and M. Dürmuth, "Will Any Password Do? Exploring Rate-Limiting on the Web," in *Who Are You?! Adventures in Authentication Workshop (WAY '18)*, pp. 1–5, Aug. 2018.

8) **T. Schnitzler**, C. Utz, F. M. Farke, C. Pöpper, and M. Dürmuth, "POSTER: User Perception and Expectations on Deleting Instant Messages – or – "What Happens If I Press This Button?"," in *USENIX Symposium On Usable Privacy and Security (SOUPS '18)*, USENIX Association, Aug. 2018.

9) F. M. Farke, L. Lorenz, **T. Schnitzler**, P. Markert, and M. Dürmuth, "You still use the password after all – Exploring FIDO2 Security Keys in a Small Company," in *USENIX Symposium On Usable Privacy and Security (SOUPS '20)*, USENIX Association, Aug. 2020.

10) C. Utz, S. Becker, **T. Schnitzler**, F. M. Farke, F. Herbert, L. Schaewitz, M. Degeling, and M. Dürmuth, "Apps Against the Spread: Privacy Implications and User Acceptance of COVID-19-Related Smartphone Apps on Three Continents," in *ACM CHI Conference on Human Factors in Computing Systems (CHI '21)*, ACM, May 2021.

11) M. Kowalewski, F. Herbert, **T. Schnitzler**, and M. Dürmuth, "Proof-of-Vax: Studying User Preferences and Perception of Covid Vaccination Certificates," in *Proceedings on Privacy Enhancing Technologies (PETS '22)*, vol. 2022, no. 1, pp. 317–338, Sciendo, Nov. 2021.

12) V. Rimmer, **T. Schnitzler**, T. Van Goethem, A. Rodríguez Romero, W. Joosen, and K. Kohls, "Trace Oddity: Methodologies for Data-Driven Traffic Analysis on Tor," in *Proceedings on Privacy Enhancing Technologies (PETS '22)*, vol. 2022, no. 3, pp. 314–335, PoPETS, Jul. 2022.

13) M. Degeling, C. Utz, F. M. Farke, F. Herbert, L. Schaewitz, M. Kowalewski, S. Becker, **T. Schnitzler**, and M. Dürmuth, "Die Nutzung von Smartphone-Apps zur Eindämmung von COVID-19 in Deutschland," in D. Krämer, J. Haltaufderheide, and J. Vollman (Ed.), "Technologien der Krise – Die Covid-19-Pandemie als Katalysator neuer Formen der Vernetzung," pp. 133–153, Transcript Verlag, Bielefeld, Germany, Jul. 2022.

14) F. Herbert, M. Kowalewski, L. Lassak, **T. Schnitzler**, and M. Dürmuth, "Fast, Easy, Convenient. Studying Adoption and Perception of Digital Covid Certificates," in *USENIX Symposium On Usable Privacy and Security (SOUPS '22)*, USENIX Association, Aug. 2022.

15) P. Markert, **T. Schnitzler**, M. Golla, and M. Dürmuth, "As soon as it's a risk, I want to require MFA: How Administrators Configure Risk-based Authentication," in *USENIX Symposium On Usable Privacy and Security (SOUPS '22)*, USENIX Association, Aug. 2022.

16) T. Heijligenberg, **T. Schnitzler**, and K. Kohls, "Station to Station: Exploring Traffic Analysis Attacks in the Mobile Domain," *under review (Int'l Workshop on Security in Mobile Technologies)*, Mar. 2022.

17) M. Kowalewski, C. Utz, M. Degeling, **T. Schnitzler**, F. Herbert, L. Schaewitz, F. M. Farke, S. Becker, and M. Dürmuth, "Attitudes Towards COVID-19 Apps Over the Course of the Pandemic," *under review (CSCW 2023)*, Jul. 2022.

1.4. Outline

The remainder of this thesis is structured as follows. Part I addresses self-published online data with particular focus on longitudinal management and revocation of such data. In Chapter 2, we provide an extensive review of previous work in this field from *technical* and *user* perspectives, systematically identify conflicts between these two sides, and derive open challenges that need to be tackled [176]. In Chapter 3, we study data revocation from the *user* perspective in a practical context, i. e., deleting messages in instant messengers [178, 179]. Chapter 4 presents a novel and fundamentally different *technical* approach for data revocation, in which owners and processors of online data formally agree on data expiration, along with penalty mechanisms in the case of violations of the agreement [175]. Part II addresses usage-driven information revelation. Chapter 5 introduces the concepts of traffic analysis, which comprises the fundamental idea behind the privacy threats in Tor and in instant messengers. In Chapter 6, we present novel attacks that enable adversaries to conduct traffic analysis, e. g., for uncovering users' identities and the services they use, more targetedly by assessing attack success chances in advance [177]. In Chapter 7, we demonstrate an attack that enables messenger users to secretly spy on the whereabouts of their contacts by sending them instant messages. We finally summarize the findings of both parts of this work in Chapter 8 and provide an outlook to possible research directions for future work.

Part I.

Managing Self-Published Online Data

2

The State of Data Revocation Research

Contents

2.1. Introduction

In their everyday life, users create huge amounts of data, shared online with varying audiences for different purposes. Whereas the initial action of sharing information is usually grounded in a deliberate decision, users mostly do not actively track the availability of their online data later on [40, 97, 129]. Hence, there is a need for continuous exposure controls and increased attention for the lifetime ending of personal online data to avoid the formation of publicly accessible information graveyards that are left unattended by users.

Topics evolving around data sovereignty have also received increased awareness due to the establishment of the *Right to be Forgotten* [242] as part of the European General Data Protection Regulation (GDPR) [60], even though data shared in online spaces is not the focus of this directive. While information processing and dissemination yet are essential aspects of privacy [194], reducing online exposure also facilitates other aspects – it can help to keep track of more important content, and fade out the rest. In the end, it can be deemed the users' sheer right to determine what is supposed to happen with their data, and for how long they prefer it to remain available.

2.1.1. Problem Statement

Compared to the non-digital past, in which forgetting information was inherent, today's world with technical capabilities to permanently store information needs actively managed processes to reduce information exposure and eventually realize forgetting [119]. In many cases, published content is not meant to be available permanently, but is only relevant for a short period of time in a certain context, e. g., when posted impulsively or out of momentum [18, 173]. Content visibility might also not match data owners' perceptions as they did not foresee sharing consequences and, therefore, requires later adjustment [190, 232]. When considered

outside their specific context, online postings can unpredictably develop dynamics that can harm users even years later [69, 172]. Thus, there is a need for individual means for users to control and adjust exposure settings.

A high-level overview of users' means to control their online exposure is provided by Bishop et al. [25]. Proposed strategies to limit the dissemination of data include proactively employing sophisticated access control mechanisms, or hiding the information within the enormous amount of data available online, e. g., by releasing large amounts of similar false information to confuse the interpreter. Additionally, research has put great efforts into developing technical approaches to assist users in managing their longitudinal privacy in general, and realizing data revocation in particular. However, such proposals have not found their way to wide-scale adoption, even though there has been a trend towards the use of tools providing better privacy and even some level of ephemerality [184].

There is evidence that users have detailed perceptions of how to share data in a wide range of contexts, but lack appropriate means to fulfill their goals. It has been shown, for a domesticity context, that users can precisely formulate who may access which of their data [120]. Moreover, users can distinguish different use cases when handling data and, therefore, switch between channels for communication and data sharing, depending on the task and content type [191]. On the downside, it also turned out that users have false perceptions of deleting data shared with others through online services [162] or in instant messengers [179].

2.1.2. Contribution

In our work, we take a closer look at the gap between how people use sharing mechanisms and privacy controls for their online data and concepts proposed by academia in order to facilitate online privacy management. To capture how people actually use online sharing mechanisms and privacy, we survey a large body of user studies carried out over the

last decade. We categorize these studies along usage patterns, drivers that make users decide to unshare or reduce the exposure of their contents, and desires they have to improve their privacy experience. On the technical side, we survey concepts and proposals that assist users in managing their longitudinal privacy and the availability of their shared online data. We categorize these proposals along the use cases they have been designed for, the adversarial models they take into account, and the underlying protection mechanisms they avail to realize their privacy features.

By evaluating our systematization, we reveal conflicts between these two sides, such as intended use cases that do not appropriately reflect actual usage patterns. Referring to such conflicts, we derive a set of challenging open problems that need to be tackled by future research in order to develop privacy-enhancing technologies that can better assist users in managing their longitudinal online privacy and the availability of their data.

Our work is the first of its kind in combining knowledge from both user studies and technical mechanisms, providing a rich understanding of research efforts on longitudinal privacy management. In summary, we provide the following contributions:

- We systematize how users interact with online services such as social networking sites in terms of their longitudinal online privacy management.
- We provide a taxonomy for technical systems to realize data revocation or to reduce exposure of publicly shared personal content as proposed in research.
- Based on the systematic analysis of previous work, we derive a set of challenges and open research questions that future research on data revocation and longitudinal privacy management should aim to tackle.

2.2. Systematization Methodology

We start systematizing existing research on longitudinal online privacy management by systematically collecting publications from major academic computer security and privacy venues or broader venues related to and relevant for our topic*. We focus our targeted paper selection on the last decade. We identified a broad range of papers based on title and abstract and decided upon adding a publication to our final set of literature after having determined its general focus by skim reading its essential sections. We further take into account cross-references starting from the resulting literature set to achieve broad academic coverage of the topic.

Given this body of literature, we study the problem of managing the availability of personal online information from two perspectives: (i) Understanding user habits and desires regarding their longitudinal online privacy and (ii) Collecting technical proposals and concepts that are designed to manage online privacy. We provide an overview of our categorization process in Figure 2.1 and describe its methodology as follows.

2.2.1. Categorization Process

The initial systematizations of the two perspectives were drafted by one author each. This included selecting the initial sets of papers, creating a first set of labels as a means to categorize these papers, and assigning each paper such labels. Subsequently, four researchers in our team thoroughly discussed the initial systematizations in several rounds. Any concerns regarding label assignments or the set of papers had to be resolved, and updates required joint agreement of all four researchers.

As we will explain in-depth in Section 2.3, we systematize research on user attitudes towards privacy management and how users perceive selected aspects of it. For each publication in the list, we provide ba-

*We focus on IEEE S&P, USENIX Security, ACM CCS, NDSS, PETS, SOUPS, and CHI.

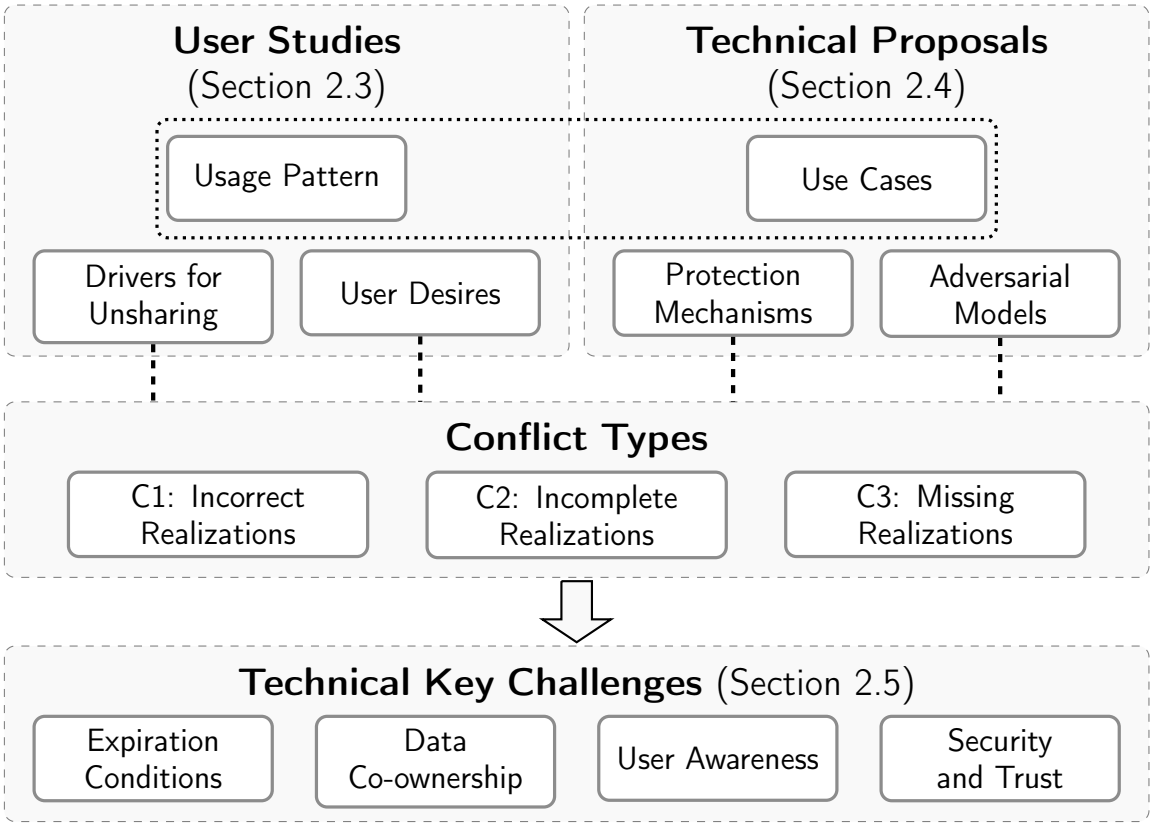

Figure 2.1. High-level overview of our systematization methodology. We categorize previous work on User Studies and Technical Proposals along a set of features. Based on the interplay among different features, we derive technical or conceptual challenges worth to be further investigated.

sic study meta-data and extract whether the work explicitly refers to longitudinal aspects of online privacy. We categorize research along the privacy management (*usage patterns*) that is covered, the identified reasons that make users change their initial privacy configuration (*drivers for unsharing*), and what *user desires* lead to a presumably improved privacy management experience.

In Section 2.4, we examine privacy controls that have been proposed or implemented as proofs-of-concepts. We systematize these controls and mechanisms along the *use cases* they have been designed for. We further categorize the *adversarial models* or adversarial settings that they should protect from, as well as the underlying *protection mechanisms* that they apply.

Discussing the set of papers was particularly necessary in the case of borderline papers, e. g., when it was unclear whether a paper indeed addressed publicly shared online data, which was a requirement for inclusion in the user studies systematization. We agreed that sharing data in cloud storage with an indefinite audience (e. g., co-students) should be sufficient to be considered *publicly* shared (cf. [97]). Similarly, for the systematization of technical proposals, detailed discussions were held when it was unclear whether a proposal limited the availability of online data. For example, we agreed that adversarial examples helped reduce shared photos' detection by smart recognition systems and therefore, these perturbations do indeed serve the users' goal of limiting availability of their online data (cf. [130]).

We further adapted the set of categories using the same process. For example, we initially considered misconceptions expressed in user studies as a separate category; they however turned out to be too diverse to be systematized in detail. We decided to focus on misconceptions that affected users' decisions about reducing exposure of their data, rendering them a sub-category of *Drivers for Unsharing*. On the technical systematization side, we decided to introduce insider adversary as a separate *adversarial model* after noticing that the existing threat models were not fully capturing the risks covered by this case.

One way to connect the two systematizations is by contrasting *usage patterns*, i. e., how users interact with privacy management options, and the *use cases* technical proposals are intended for, i. e., what they offer users for managing their privacy. Both systematizations capture to what extent content exposure can be limited or entirely ended, and if there is active user interaction involved in this process.

2.2.2. Deriving Challenges

Starting from the categories identified in either part of the systematization, we identified potential inconsistencies or conflicts between them.

Pursuing a user-centric approach, we systematically examined to what extent users' desires and their drivers for unsharing are reflected in the current state of technical proposals. We identified conflicts, whenever realizations in technical proposals are (i) *incorrect*, i. e., orthogonal to users' needs, (ii) *incomplete*, i. e., promising but far from satisfying users' requirements, or (iii) *missing*, i. e., not addressing users' desires at all. For each conflict, we derived challenges on how such inconsistencies can be addressed.

By combining and contrasting knowledge from both of the obtained systematizations, conflicts were identified and challenges were derived by two researchers individually first and then discussed and iteratively updated. Again, challenges were subject to discussions among four researchers – proposals and concerns brought up by anyone of them had to be resolved and any updates required agreement of all four researchers.

As we will detail in Section 2.5, we followed a bottom-up approach: first, we derived fine-grained challenges related to conflicts, and then we put them into a broader context and related them to each other, resulting in a set of four challenge groups. The challenges we identify refer to (i) the *expiration conditions* under which data are supposed to disappear, (ii) *user awareness* of how particular privacy controls actually work, (iii) multi-user conflicts, which originate in the implicit *co-ownership of data*, when data affects the privacy of more than one individual, and (iv) issues regarding *security and trust* w. r. t. specific actors users consider when making changes in their online exposure.

2.3. Categorizing User Interaction

We first systematize users' preferences and behavior w. r. t. their longitudinal online privacy. We explain the different categories in our taxonomy and summarize our findings in Table 2.1. We arrange publications in three groups, each of which is ordered chronologically with most recent publications first.

Table 2.1. Systematization of User Studies on Longitudinal Online Privacy. We arrange surveyed publications in three groups, (i) papers explicitly referring to *longitudinal* aspects of privacy, (ii) papers that study publicly shared data without referring to longitudinality, and (iii) papers that are still relevant to the topic but do not cover any of the categories we present in our systematization. Publications within each groups are ranked in chronological order (most recent publications first).

Publication		Study Data					Usage Pattern							Drivers for Unsharing						User Desires				
Reference	Venue	Study Type	Platform	Sample Size	Participants Sample	Female/Male [%]	Publicly Shared Data	Longitudinal Data	Delete Content	Delete Account	Reduce Exposure (Act.)	Reduce Exposure (Pass.)	Auto-expire	Irrelevance	Change of Opinions	Regrets	Events	Misconceptions	Fears	Reduce Visibility (Time)	Content-based Audience	Control Friends' Content	Confirm Delete	User-view
[129]	CCS'19	R	FB	78	AMT	69/31	●	●	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○
[125]	SOUPS'18	S	-	30	UNI	60/40	●	●	○	○	○	○	●	●	○	○	○	○	○	●	○	○	●	○
[135]	SOUPS'18	S	-	22	–	50/50	●	●	●	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○
[97]	CHI'18	R	CL	100	AMT	41/59	◐	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○
[128]	J-IEEE-IC'17	P	TW	100K	[P]	-	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○
[11]	J-HCI'17	S	FB	272	AMT	61/38	●	●	○	○	○	○	○	●	○	○	●	○	○	●	●	○	○	○
[127]	SOUPS'16	P	TW	100K	[P]	-	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○
[17]	WPES'13	R	FB	299	AMT	55/44	●	●	○	○	○	○	○	●	●	○	●	●	○	○	○	○	○	○
[10]	SOUPS'13	S	FB	193	AMT	40/59	●	●	○	○	○	○	○	●	○	○	●	○	○	●	●	○	○	○
[7]	SOUPS'19	S	FI	30	CON	50/50	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○
[73]	CHI'19	S	SC	1515	Q	57/43	●	○	○	○	○	○	●	○	○	○	○	○	○	●	○	●	○	○	○
[163]	SOUPS'18	S	-	23	UNI	52/48	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○
[173]	CHI'17	S	YK	18	UNI	56/44	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○
[252]	WWW'16	P	TW	30K	[P]	-	●	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○
[24]	WLSM'16	P	TW	203K	[P]	-	●	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○
[51]	J-CHB'15	S	FB	380	CON	52/45	●	○	○	○	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○
[112]	WLSM'14	P	TW	ALL	[P]	-	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[190]	CHI'13	S	TW	1221	AMT	53/46	●	○	○	○	○	○	○	○	●	●	●	●	○	○	○	○	○	○	○
[139]	HICCS'13	R	FB	68	UNI	38/62	●	○	○	○	○	○	○	○	●	○	○	●	○	●	○	○	○	○	○
[6]	CSCW'13	P	TW	292K	[P]	-	●	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○
[117]	PerCom'12	S	FB	65	UNI	62/38	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	○	○	○	○
[90]	SOUPS'12	R	FB	260	WEB	75/25	●	○	●	○	●	●	○	○	○	○	○	○	●	○	○	○	○	○	○
[232]	SOUPS'11	S	FB	569	AMT	64/36	●	○	○	○	○	○	○	○	●	●	○	●	●	○	●	○	○	○	○
[56]	CHI'11	E	FB	33	UNI	50/50	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[167]	IFIP-HCI'11	P, S	FB	103	WEB	59/41	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[23]	CHI'10	S	FB	14	UNI	57/43	●	○	○	○	○	●	○	○	●	●	○	○	●	○	○	●	○	○	○
[110]	UPSEC'8	E	FB	16	UNI	44/56	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●
[44]	PETS'17	S	-	60	AMT	37/63	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[63]	CSCW'17	R	FB	1706	AMT	58/41	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[191]	CHI'16	S	-	17	WEB	65/35	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[126]	SOUPS'14	R	FB	1239	WEB	24/76	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[206]	JPC'13	P	FB	5076	[P]	-	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
[111]	IMC'11	S	FB	200	AMT	46/54	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Study Type *S*: Self-reported data, *P*: Public data analysis, *E*: Experiment based on prototype implementations, *R*: Survey with real user data **Platform** *TW*: Twitter, *FB*: Facebook, *SC*: Snapchat, *CL*: Cloud Storage, *YK*: Yik Yak, *FI*: Fitness Social Networks **Participants Sample** *AMT*: Amazon Mechanical Turk, *Q*: Qualtrics, *WEB*: Other Web Platforms, *UNI*: University Sample (various recruiting methods), *CON*: Convenience Sampling (Offline), *[P]*: Public data analysis –: No info provided

2.3.1. Study Data

For each piece of research we cover in our systematization, we report the type of the user study that has been conducted: Self-reported data (S), Exploring real-world data with self-reported answers (R), Experiments based on prototype implementations (E), or Analyzing publicly available data sets (P).

Most studies cover scenarios that reflect a situation on a particular online platform, sometimes with a very specific focus, such as fitness social networks. While most studies have covered Facebook (FB) and Twitter (TW), we also find research on Snapchat (SC), Cloud Storage (CL) provided by Dropbox and Google, Fitness (FI) social networking sites, and the subsequently shut down platform Yik Yak (YK).

We further denote the number of participants that have taken part in each study (*Sample Size*), how participants have been recruited (*Participants Sample*), and basic demographics in terms of a gender distribution to provide information about the meaningfulness of results.

Considering the study type and the participants sample can usually hint towards potential study limitations. Qualitative research typically studies significantly smaller sample sizes, thus providing detailed insights into very specific issues, compared to quantitative studies having larger groups of participants. However, even large samples, e. g., recruited via Amazon Mechanical Turk, do not always generalize for all users of a specific platform under observation, not at all for users of other platforms. Furthermore, it must be considered that self-reported data may not be as meaningful as practical experiments with real user content since alleged privacy attitudes have been shown to differ from actual behavior [44]. On the downside, practical experiments with real user data may deter rather privacy-sensitive users from participating in the study [129].

The focus of our systematization is on studies that explore *Publicly Shared Data* (denoted with ● in the respective column), which applies to all but one study [97] that partially covers public data (◐) since it

primarily focuses on data stored in the cloud that *can* be shared with a limited audience. In a similar fashion, we also denote whether a study explicitly refers to longitudinal aspects of data sharing (●) or not (○).

2.3.2. Usage Patterns

We extract a set of *Usage Patterns* that can be applied to limit the exposure of online content, ranging from explicit deletion operations to exposure reduction, and auto-expiry. We define the patterns we identified within the existing literature as follows:

- *Delete Content* is an explicit action performed by a user to entirely remove content from a platform.
- *Delete Account* is another explicit action performed by users that entirely removes all of their content from the platform and also their account, such that there remains no direct representation of them on that platform.
- *Reduce Exposure (Actively)* covers controls users apply to actively manage the audience for a piece of content, such as, e. g., changing its visibility settings from public to friends only.
- *Reduce Exposure (Passively)* captures features that remove references from exposed content, without actually altering the content availability, such as, e. g., un-tagging a specific person in a shared photo.
- *Auto-expire* covers all mechanisms ensuring that published contents are made unavailable automatically when certain conditions are met. In particular, expiration takes effect without any further action to be taken by the owner or publisher of the content after its initial publication.

Previous work studies one or more of these patterns in detail within specific application scenarios. In Table 2.1, we mark this with a filled circle (●). If the usage pattern is not covered by a paper, we denote this with an empty circle (○).

2.3.3. Drivers for Unsharing

When it comes to the end of data lifetime, we are interested in users' motivation behind their decision to limit the visibility of data. We identified several drivers that determine users to unshare content on online platforms:

- *Irrelevance* denotes a situation in which content is withdrawn because it has become irrelevant or unimportant for the owner or its audience, and there is no more reason to keep it online.
- *Change of Opinions* indicates that content is withdrawn since the owner changed their opinion about the content exposure, without further specifying reasons.
- *Regrets* captures situations in which users revised their decisions to publish content due to explicitly stated regrets that came up after publication.
- *Events* means that some external event unrelated to the initial publishing has made its owner reason differently about the current level of exposure.
- *Misconceptions* denotes a general term that applies when participants expressed the actual level of exposure does not match what they perceived. In case there is a misconception, other factors (e. g., oversharing) may simultaneously apply.
- *Fears* captures situations in which users stated that they feared that specific groups of people could see their contents.

For all these features, we mark whether they were referred to in the considered publications (●) or they were not covered (○).

2.3.4. User Desires

In several studies, users have expressed desires for features facilitating their interaction with online services. Whenever such a desire is related to longitudinal online privacy or managing their online exposure, we

consider it in our systematization. We identified five related user desires in our literature set:

- *Reduce Visibility (Time)* indicates that users expressed data to become less exposed over time after being published.
- *Content-based Audience* covers cases in which users desired to have the audience composed differently depending on the content of the data being published.
- *Control Friends' Content* means that users desired to control contents owned by their friends (in cases it affected their privacy).
- *Confirm Delete* captures cases in which users expressed that they did not want to have data automatically disappear, but preferred being prompted to confirm its deletion.
- *User-view* denotes a desired feature where users can view their own profile from the perspective of another user to better estimate the specific exposure implications of their privacy configuration.

2.4. Categorizing Technical Proposals

Technical proposals to tackle longitudinal privacy concerns have been considered and developed for a variety of platforms, such as online social networks (SN) like Facebook (FB) and Twitter (TW), cloud-based applications (CL), and messaging applications (MA); we also consider proposals that are platform-independent (PI). For the systematization of the technical proposals, we consider the *use case* for which they were designed, the *adversarial assumptions* under which they operate, and the *underlying protection mechanisms* they rely upon. We summarize our findings in Table 2.2 that arranges proposals in a chronological order with most recent publications first.

Table 2.2. Systematization of Technical Proposals for Longitudinal Online Privacy. We arrange surveyed mechanisms designed for a variety of platforms, use cases, adversarial assumptions and underlying protection mechanisms. Publications are ranked in a chronological order with most recent publications first.

Publication			Use Cases				Adversarial Models				Underlying Protection Mechanisms						
Reference	Venue	Platform	Delete Content	Reduce Exposure	User Involvement	# of Data Owners	Retroactive	Honest-but-curious	Interfering	Insider	Cryptographic/Signatures	Distributed Architecture	Adversarial Examples	Deception & Flooding	Access Control Policies	Game-theoretical	Others/[Specifics]
[123]	PETS'19	TW	○	●	P	1	○	○	●	○	○	○	○	●	○	○	Intermittent withdrawal
[246]	ForensicSec'19	CL	○	●	P	n	○	●	●	○	●	○	○	○	●	○	[Attribute-based collaboration]
[175]	IFIP-SEC'19	PI	●	●	P	1	●	○	○	○	○	○	○	○	○	○	Smart contracts
[68]	NeurIPS'19	PI	●	○	P	1	●	○	○	○	○	○	○	○	○	○	Quantized k-means
[146]	NDSS'18	PI	○	●	A	n	○	●	●	○	●	○	●	○	●	○	Identity management system
[8]	CODASPY'18	PI	●	○	P	1	○	●	●	○	●	●	○	○	○	○	[Time-lock puzzles]
[85]	CODASPY'17	SN	○	●	P	n	○	●	○	●	●	●	○	○	●	○	[Threshold secret-sharing]
[145]	ICCV'17	SN	○	●	P	1	○	●	●	○	○	○	●	○	○	●	[Adversarial Image perturbations]
[130]	CVPR'17	SN	○	●	P	1	○	●	●	○	○	○	●	○	○	○	[Adversarial Image perturbations]
[161]	GameSec'17	SN	○	●	P	n	○	●	●	○	○	○	○	○	●	●	[Negotiation]
[235]	ETHReport'17	CL	●	○	A	n	○	●	●	●	●	●	○	○	○	○	[Group secret]
[12]	CCS'16	CL	●	●	A	1	●	●	○	○	●	○	○	○	○	○	Interdependency in encrypted
[250]	CODASPY'16	PI	●	○	P	1	●	○	○	○	●	●	○	○	○	○	[DNS Caching]
[207]	TKDE'16	SN	○	●	P	n	○	●	○	○	○	○	○	○	●	○	[Computational conflict resolution]
[33]	S&P'15	PI	●	○	P	1	●	●	●	○	○	○	○	○	○	○	Machine Unlearning
[141]	SIGMOD'15	PI	●	●	P	1	○	●	●	○	○	○	○	○	○	○	Brain-inspired data retention
[2]	ACM-SCC'15	CL	●	○	P	1	○	●	●	○	○	○	○	○	○	○	Forgetful data structures
[193]	CCSW'13	CL	○	●	P	1	○	●	●	○	●	○	○	○	○	○	Heterogeneous documents
[25]	NSPW'13	PI	○	●	A	1	○	●	○	●	○	○	●	●	○	○	[False attribution]
[205]	IEEE-PST'13	SN	○	●	A	n	●	●	○	○	●	●	○	○	●	○	User-to-content relations
[50]	S&P'12	TW	○	●	P	1	○	●	●	○	●	○	○	○	○	○	[Blind RSA signatures]
[166]	WPES'12	PI	●	○	P	1	●	●	○	○	●	●	○	○	○	○	Statistical webpage changes
[19]	PETS'11	SN	○	●	A	1	○	●	●	○	●	○	○	○	●	○	[OpenPGP]
[35]	ICNP'11	PI	●	○	P	1	●	○	○	○	●	●	○	○	○	○	[DNS Caching]
[66]	UW-CSE'11	PI	●	○	P	1	●	●	○	○	●	●	○	○	○	○	Integrating diverse mechanisms
[34]	CollbCom'11	SN	○	●	P	n	○	●	○	○	○	○	○	○	●	○	[Aggregation of policies]
[216]	PETS'10	SN	○	●	P	n	○	●	○	○	○	○	○	○	●	○	[Aggregation of policies]
[23]	CHI'10	FB	○	●	A	n	○	○	●	○	○	○	○	○	●	○	[Manual conflict resolution]
[241]	POLICY'10	SN	○	●	A	n	○	●	○	○	○	○	○	○	●	○	[Manual conflict resolution]
[154]	ACSAC'10	MA	●	○	P	1	○	○	●	●	●	○	○	○	○	○	Porter storage
[65]	USENIX'09	PI	●	○	P	1	●	○	○	○	●	●	○	○	○	○	[DHTs of P2P networks]
[195]	WWW'09	SN	○	●	P	n	○	●	○	○	○	○	○	○	●	●	Auction-based inference
[115]	CSE'09	SN	○	●	P	1	○	●	●	○	●	○	○	●	○	○	Third party storage server
[27]	SecureCom'09	PI	○	●	A	1	○	○	●	●	○	○	○	●	○	○	Bait information
[152]	SMLI'05	MA	●	○	P	1	●	○	●	○	●	○	○	○	○	○	[Centralized server storing keys]

Platform *TW*: Twitter, *FB*: Facebook, *SN*: (general) Social Networks, *CL*: Cloud Storage, *MA*: Messaging Applications, *PI*: Platform Independent
User Involvement *A*: Active, *P*: Passive
# of data owners *1*: Single user scenario, *n*: Multi-user scenario

2.4.1. Use Cases

For each technical proposal we cover in our systematization, we detail the functionality it is intended to serve:

- *Delete Content* results in removing a piece of content from a platform so that it is no longer publicly accessible. A proposal that provides such guarantees is labeled ●, as opposed to ○.
- *Reduce Exposure* allows users to manage the visibility of a piece of content on a platform such that it is exposed only to a subset of the previous audience. A proposal that allows such functionality is labeled ●, as opposed to ○.
- *User Involvement* captures the nature of the involvement of the data owner while limiting content availability. If the process requires the data owner to actively change the content availability, it is labeled active (A). Otherwise, if the process relies on a mechanism that ensures automatic change in the availability of published content, then we denote it as passive (P). The passive case turns out to be more common.
- *# of Data Owners* captures the number of users making the decision to change the availability of content. In most cases, the data is owned and uploaded by a single user, denoted by 1. Multi-user scenarios that involve content co-owned by more than one user are denoted by n and are also common, but apply to slightly fewer proposals.

2.4.2. Adversarial Models

The Dolev-Yao (DY) adversary model is widely used to analyze system and network protocols [36]. For many settings, this model is, however, too strong: many legitimate participants of the protocol, such as service providers or fellow users with varying degrees of association, do not qualify to be DY adversaries. This does not imply that these parties cannot be malicious, though, so it is important to consider the relevant

threat vectors. We, therefore, analyze the privacy guarantees of existing proposals against the following threat models:

- *Retroactive adversaries* learn which data they are interested in only after the data has been revoked/expired. This threat model makes an assumption that the attacker has no interest in accessing the published data prior to its expiration. Since the data was publicly available during its lifetime, it is not assumed to be private and accessible by everyone. However, past its expiration time, the privacy of deleted data is ensured.
- *Honest-but-curious adversaries* act as a legitimate party in a protocol that will not deviate from the definition but will attempt to learn as much information as possible. The majority of these adversaries are service providers who are handling users' data and running analyses on top of it. These adversaries are also referred to as 'curious-but-non-interfering' or 'passive' mainly due to their tendency to indiscriminately collect data once available in the hope that it may be of interest to them in the future.
- *Interfering adversaries* actively interfere with the private information of the user, either preponing or postponing the event limiting the availability of the content. This threat model treats clients in the system as untrusted: they may bypass the system to publish sensitive content without obtaining consent from the target users through means such as colluding with other malicious clients and deviating from the protocol description.
- *Insider adversaries* control user devices, including porter devices, and can compromise users' passwords and passphrases. An insider attack may be intentional or accidental. Insider attackers range from poorly trained administrators who make mistakes, to malicious individuals who intentionally compromise the security of systems.

We rate the adversarial model of each technical proposal w. r. t. these attacker types. If a proposal considers a specific adversary in their threat

model, we label it with ●. Otherwise, if it provides no guarantees against a specific adversary, then it is labeled with ○. The honest-but-curious adversary is the most commonly considered threat model, but the other adversaries are also being considered when technical solutions are proposed.

2.4.3. Underlying Protection Mechanisms

To realize use cases and fulfill adversarial guarantees, each proposal relies on different technical mechanisms. A number of protection mechanism principles have been proposed multiple times in varying realizations; others have occurred less frequently.

- *Cryptographic* mechanisms embed encryption keys into stored data within centralized or distributed storage systems. They may control the extent of the keys' replication to prevent the key from being recovered from the underlying storage after a configurable amount of time. Most of the time-based data revocation proposals rely on encryption by uploading the data in encrypted form along with information on where and how to gather the decryption key during content's lifetime. This category also covers *digital signatures* that allow users to embed signatures to the content.
- *Distributed Architectures* allow members to collectively generate and distribute group secrets among themselves. In order to avoid single-point failures, cryptography-based forgetting schemes avoid putting trust in a central authority for the storage of keys [35, 65]. Instead, they rely on key-sharing and distributing parts of the decryption key on distributed storage. Some approaches have yielded support for an 'expiration date' of a few days by spreading bits of the key among random indices in the DHT [65] whereas others demonstrated expiration times of up to months by exploiting the evolving nature of webpages and using threshold secret sharing scheme to reconstruct the key [166].

- *Adversarial Examples* confuse AI/recognition systems effectively by generating additive perturbations that are invisible to the human eye, thus without introducing unpleasant artifacts. Given the prevalence of AI systems, such as facial recognition, adversarial examples could allow users to limit their content's exposure to these algorithms (i. e., go undetected.)
- *Deception & Flooding* approaches require the subject to release large amounts of similar synthetic, but convincing, information that is not correct. The viewer is thus challenged to pick the correct confidential information from the mass of incorrect information.
- *Access Control Policies* are the classical approach to specify how access is managed and who may access information under what circumstances. These policies can be set manually, computed through aggregation, or learned over time using ML algorithms.
- *Game-theoretical* frameworks aim to achieve optimal decision making of independent and competing actors in a strategic setting. It can be used to understand and predict the effect of multi-party involvement in access control decisions on individual behaviors of social network users.
- *Others/[Specifics]:* In addition to the above categories, the existing literature relied on less-frequent protection mechanisms, such as approaches that mimic the human brain, smart contracts, porter storage devices, etc. We list them individually by name. In some cases, we also list specifics of mechanisms covered in one of the above categories. In such a case, we list them in brackets, for it is an explanation instead of a new category.

2.5. Technical Key Challenges

Based on our systematizations in Sections 2.3 and 2.4, we determine a set of technically challenging problems that have not been solved to date. We explore to what extent users' desires and their drivers for

unsharing, as expressed in user studies, have been realized as part of technical proposals. Whenever we identify factors that have not been appropriately addressed on the technical side, i. e., when realizations are incorrect, incomplete, or missing, we identify this as a conflict to be resolved, each resulting in one or more challenges.

We determine these challenges first and then group similar ones and consider them also in context with each other. Our systematization results in challenges that are broadly categorized regarding (i) the expiration conditions under which data are supposed to be rendered unavailable (Section 2.5.1), (ii) the co-ownership of data resulting in potential conflicts among multiple users (Section 2.5.2), (iii) user awareness regarding the functionality of privacy controls (Section 2.5.3), and (iv) security and trust relations among the parties involved in data publishing (Section 2.5.4). The overall list of challenges per group is illustrated in Figure 2.2.

2.5.1. Expiration Conditions

Multiple studies reported in Section 2.3 have found that participants did not want contents to fade away wholesale with age [11, 17, 97]. Whereas participants of these studies have shown a preference for a handful of posts to become more private over time, they demonstrated their desire to make some posts *more visible* over time. Thus, the decision on content's exposure control is a complicated one, hardly captured in the true sense by focusing alone on the age of posting.

Studies have identified other contextual factors such as inactivity of the post (e. g., lack of viewing/sharing) [127, 128] and major life events (e. g., moving to a new city or graduation) [10] that could impact users' desire to keep the data publicly available. Users' preference to limit exposure also largely depends on the content of their data, and effective audience control mechanisms can facilitate their openness to share [11,120,135]. In this regard, private-by-default interfaces, such as Snapchat, that allow

Figure 2.2. Overview of the challenges we derived from conflicts identified in the systematizations of user studies and technical proposals, grouped by four topic areas: Expiration Conditions, Data Co-ownership, User Awareness, and Security and Trust. We denote to which feature(s) of the user studies systematization each challenge refers (bottom line) and to what extent they are currently addressed in technical proposals (in terms of realization level).

audience-related considerations to be made on a per-post basis, result in users being much more audience-aware [4, 73]. In contrast, content sharing interfaces that are not as intuitive to per-post based audience decisions result in content being overexposed w. r. t. the uploaders' intentions [21].

The overview of technical proposals in Section 2.4 shows that the most commonly considered condition for data revocation in previous academic proposals is the time passed since publication [65,152,166]. Solutions for end-users also use time as an expiration condition [147, 157, 192, 214]. Time-based mechanisms for data revocation are easily comprehensible and provide transparently decidable expiration conditions. However, each expiration time is determined and set at the time of publishing of data, which leads to a three-fold conflict:

(i) the appropriate time for data revocation is often difficult to determine in advance,
(ii) the context in which data is published (and in which the expiration condition is set) can change, which may require to adapt the expiration condition, and
(iii) no context information or other potentially relevant aspects for deciding whether data should remain online or not are taken into consideration when the expiration condition is determined.

Improving revocation mechanisms is a complex problem, as it must take into account multiple contradictory factors, such as the desire to retain some old content while allowing other content to be completely removed. Based on our systematization of user studies and technical proposals, we identify and present challenging research dimensions that are desired by the users but have not yet been effectively realized in the technical implementations.

The first two challenges, A-1 and A-2, tackle missing realizations, taking into account multiple drivers for unsharing as expressed by users. Challenge A-3 takes up on work that already considers relevance as a factor to determine expiration, focusing on how to overcome its yet in-

complete realization. We emphasize that there is an overlap between A-1 and the two subsequent challenges. Whereas A-1 provides a more holistic viewpoint, the other two can be considered specific cases of it. However, A-2 and A-3 can also be tackled independently and do not require A-1 to be resolved. Finally, challenges A-4 and A-5 deal with incomplete and missing realizations in the interplay between published contents and audiences.

Challenge A-1: Brain-inspired Expiration

All existing mechanisms proposed have in common that the data revocation mechanism is implemented as a feature in terms of an explicit process. In contrast, Müller and Pilzecker's classical work [131] on retroactive inhibition in human memory found that forgetting is not a process that is actively triggered, but an implicit result of multiple information interfering with each other with more relevant information suppressing other information. What gets preserved in long-term memory may depend on multiple factors, including the 'meaningfulness' of the memory [28]. This can be transferred to our observations in the user studies systematization, where also multiple different factors implicitly contribute to the appropriateness of expiration conditions.

The technical challenge here is to imitate this behavior within a file storage system, i. e., to make access to information more difficult, the more new information is added, thus, waiving the need for explicitly revoking such information. In recent years, some research efforts have provided a promising start towards formalizing models imitating workings of human memory for their information management processes [2,140,141]. That being said, we are far from letting go of hard demarcation of data availability and realizing mechanisms that have contents fade away over time, which is why we keep labelling this challenge as missing (cf. Figure 2.2).

Challenge A-2: Context-based Expiration

External factors, such as changes in life circumstances, can impact users' privacy preferences for online content, possibly due to changes in social circles or individual preferences. Since users do not explicitly formulate contextual factors, such as major life events, reflecting them in the deletion mechanism is still a major technical challenge. Service providers who aggregate a lot of information about individual users would possibly be able to design mechanisms that incorporate information about users and their social circles to change the visibility of published data. However, this is rather difficult for cryptographic erasure mechanisms applied to standalone information that is published anonymously and/or not related to any other source of information. Besides its limited technical feasibility, additional information aggregation also raises questions about privacy implications.

Challenge A-3: Inactivity-based Expiration

Some mechanisms [128, 250] have attempted, with varying levels of success, to realize expiration based on the amount of attention/interactions attracted by the data object. However, sole reliance on this model does not fully capture all practical aspects: some users choose to keep/archive some content even after it becomes inactive. Thus, it is technically challenging to realize an inactivity-based expiration solution that is equipped to identify user-specific content features which contribute to their willingness to keep the content alive despite its inactive status. Another challenging aspect under such implementations is that posts containing controversial content will elicit considerable attention and thus will continue to remain in the public domain for longer.

Challenge A-4: Audience-based Expiration

People do share not only different types of data but also have multiple heterogeneous groups of audiences accessing their contents. While cryp-

tographic erasure mechanisms assume that everyone can read published data under the same conditions, there is a variety of access control settings available in social networks or cloud storage systems to satisfy the need to manage data for different target audiences. Adoption of audience-specific privacy controls suggests that not all readers of ephemeral data should be affected by exposure control decisions in the same way, but that there should be different conditions for individual users or groups of users. This leads to the technical challenge of realizing mechanisms that implement audience-dependent expiration conditions.

Challenge A-5: Content-based Expiration

Studies on changes in users' preferences about data availability have also captured the contents of data [11, 120, 135]. The challenge to realize more sophisticated expiration conditions is not limited to incorporating appropriate external factors. The data items themselves should also be taken into account, both in terms of their file formats and their contents or structural parameters. This requires to determine appropriate conditions for each type of data and to analyze data upon publishing in order to map them according to the categorization.

2.5.2. Data Co-ownership

A significant number of items uploaded to Online Social Networks (OSNs) involve multiple parties who are supposed to be interested in controlling its exposure to the public. Such items range from photos that depict multiple users to comments that mention multiple users to events in which multiple users are invited. Existing implementations of OSNs have not successfully tackled the problem of conflicting privacy preferences among users that co-own a piece of data.

In real-world applications such as Instagram, users uploading a photo can tag other users who are also present in or related to that photo. The tagged user can then control the visibility of the photo on their profile

by hiding the tagged photo or deleting the tag itself. Neither of these options affects the visibility of the tagged photo on the whole platform since followers of the uploader are guaranteed access regardless of other tagged users' visibility preferences. When we recall that even preferences of individual users do not remain constant, it appears reasonable that merging the privacy preferences of multiple users is likely to end in conflict. The lack of appropriate conflict resolution mechanisms in the current implementations of OSNs can lead to privacy violations with serious outcomes for the parties involved.

User studies on online privacy management often refer to multi-user scenarios as a use case, for example, for photos being taken at parties or social events. However, the set of research that actually covers multi-user scenarios and their implications is rather small, even though users have expressed a desire to control their friends' content when it affects them already ten years ago [23]. The only privacy management measure suitable in multi-user scenarios that is covered by several studies is untagging but from different perspectives such as its overall prevalence [51], or revisiting initially set and possibly erroneous privacy settings [90,117]. Eventually, users' strategy to overcome the risk of being unintentionally exposed publicly is preventing photos from being taken at all [163].

Research proposals that require users to collectively solve their privacy conflicts [195, 241] comprise promising concepts but lack practical evaluations of their acceptance in real-world applications. Other proposed mechanisms that automate this process rely heavily on fixed rules (majority voting, veto voting, etc.) [34, 216], thus, resulting in oversimplification of the conflict resolution process and mismatch between actual user behavior and the suggested method for resolving privacy conflicts. Such and Criado [207] proposed a promising computational model that adapts conflict resolution strategy based on the sensitivity of the item being shared and relative importance of the conflict (estimated through the strength of the relationship between owners and the target audiences). However, their mechanism does not take into account the strength of the

relationship between negotiators and the role of history of previous negotiations on concessions in the current conflict. Furthermore, the approach does not take into account the effect of types of data items under consideration. In a rather restrictive proposal by Olteanu et al. [146], photos can only be uploaded to a social network site with all faces detected in it being removed, only allowing to display them after the corresponding person has explicitly agreed.

Designing a model that is complex enough to emulate user behavior most of the time, and that requires minimum intervention from the user's side is indeed challenging. From a legal perspective, proposals that use, e. g., *majority voting* do not seem to uphold users' right to be forgotten as prescribed in the recent regulations – as soon as one of the involved users wants an item to be deleted, it has to be removed if we strictly interpret the European GDPR [153].

While multiple or evolving drivers for unsharing already apply to single-user scenarios [10, 11] (cf. Section 2.5.1), expanding their concepts to multi-user settings raises additional challenges. The challenges listed here are related to realizations of users' desires to control their friends' contents in case it also affects themselves.

Challenge B-1: Adaptability

It is technically challenging to devise a model that takes into account the past history of negotiations between co-owners when deciding on the privacy preferences for new items. Since major OSNs keep a record of all postings on one's profile, it is likely that exposure settings for the past co-owned postings may no longer serve users' privacy requirements in the present context. Individual preferences for existing items may equally evolve and need to be adapted. Allowing users the option to re-negotiate the privacy settings for co-owned items might be necessary for these models to be widely adopted. However, realizations of adapt-

able exposure controls for co-owned data items are missing in current realizations.

Challenge B-2: Handling Power Imbalance

Another challenge involving co-ownership of data on *OSNs* is that users' attitudes towards each other do not remain constant. On most of the platforms, users have the option to unfriend or even 'block' other users, rendering their profiles inaccessible. In the aftermath of such an event, users are denied the power to access the co-owned data items on the other user's profile. It is challenging to come up with a solution that honors users' unfriending decision while still ensuring their right to manage the co-owned data items.

2.5.3. User Awareness

Kang et al. identified that people with more articulated technical models on average expressed higher awareness of who could access their data [95]. A Better understanding of the number of privacy threats was found to be correlated with the protective actions taken by the individuals [151]. Internet users have been found to struggle to update their existing models at a rate comparable to the change in the internet and online platforms. In fact, prior privacy studies have identified that only a few participants expressed awareness that their models might be outdated [95]. Prior work has also called for serious attention towards the presence of age gap in information behavior. Yong found out that older people are less skillful in privacy control and, therefore, are more susceptible to become the victims of privacy-related breaches [151]. The situation is further complicated by a lack of enthusiasm on older users' part in seeking help with privacy-related technology to avoid social embarrassment. To put the demographics into perspective, Facebook alone has at least 20% of its user base aged above 45 [200]. The matters are worsened as technical mechanisms operate under various levels of adversarial assumptions and

rely on a variety of different protection mechanisms; the average user is usually not technically proficient or aware to update their mental model about different security functionalities. It is, therefore, not surprising that multiple studies reported misconceptions as one of the major drivers behind users' unsharing of data [17, 24, 190].

There also exist vast differences in the implementation of security-related features across different services (e. g. social networks vs. messaging applications) and different platforms within a service (e. g. Facebook vs. Twitter). Talking specifically about implementations of content deletion, there exist inconsistencies across:

(i) services – the way Facebook (SN) implements deletion for shared postings within a group is different from the way Facebook Messenger (MA) tackles deletion of messages in a group. Similarly, users lack information on how deletion would work for cloud storage. Findings of Ramokapane et al. study attribute users' failure to delete from cloud storage to the lack of information about how cloud and deletion within the cloud functions [162].

(ii) platforms – whereas deletion of a post on Facebook (SN) makes the related comments and re-shares on the post unavailable, it is not the same for Twitter (SN), where residual tweets (interactions associated with the withdrawn post) continue to leak information about the withdrawn tweet [127]. Similarly, disparities in the implementation of deletion functionality exist for messaging platforms. Skype (MA) allows the message sender to delete messages from the logs of all participants in the conversation, whereas Facebook messenger (MA) allows the sender to delete messages from their own conversation history only [179].

The challenges C-1 and C-2 below relate to the missing realizations taking into account drivers for unsharing (misconceptions) and desires (user view) reported by users, and inconsistencies in implementations.

Challenge C-1: Sophistication of Technical Mental Models

Users are known to formulate their own incorrect mental models when they are faced with a task to complete with their limited knowledge [234]. Given large differences in users' technical understanding and variation among mechanisms' promised adversarial guarantees, the technical challenge here is to work within existing mental models to make actual functions clearer and communicate complex privacy issues to regular users in an intuitive and correct way. Since service providers make regular changes to their interfaces and features, it is important and challenging to simultaneously update the knowledge of the end-users, to minimize the risks associated with outdated mental models.

Challenge C-2: Borrowed Mental Models

Any given internet user is likely to be a member of multiple online services as well as platforms within those services. Some users naively transfer their mental models from one platform to another. These borrowed mental models considerably hinder the correct understanding of features and can expose users' data to unintended audiences. The technical challenge is the design of user interfaces, tutorials, and control setting pages that effectively convey the consequences of different actions taken by users on a specific platform.

2.5.4. Security and Trust

The process of making data available online typically involves multiple parties interacting with the data, such as friends or contacts in social networks, service providers, advertising companies aggregating individual user profiles for marketing purposes, or other third parties proactively crawling all available web contents. Such activities are usually carried out as soon as pieces of data appear online. In contrast, the common security model used in research proposals on automated data revocation is

security against a retrospective adversary [35,65,152,166,205,250]. Basically, this type of attacker is not interested in tampering with published data during its lifetime, but only after its expiration.

In the same way, a large body of proposals rely on distributed architectures to realize expiration since centralized service providers are considered untrusted [8,35,65,85,205,250]. As a particular flaw, all types of entities are considered equally, and there are no differences between types of audiences. This is not in line with users publishing photos on platforms of large companies such as Facebook, who rather express fears such as specific groups of people (e. g. their parents or other family members) seeing their content and considering it inappropriate [163,232].

Data deletion in artificial intelligence environments is a complicated task and poses a serious threat to longitudinal aspects of users' privacy. Legal scholars have questioned the legality of using of AI systems trained on deleted data in the context of the Right to be Forgotten [224]. In fact, model inversion and membership inference attacks have already demonstrated that the information used in training a model could be reconstructed afterwards by an adversary [222]. Our systematization of technical proposals identified that few of them enable control over the availability of data that is fed into machine learning models.

In light of the failure of the existing (theoretical) adversary models to capture the actual security requirements reported by users through drivers for unsharing (Fears) and desires (Content-based Audience), challenge D-1 brings attention to incorrect realizations of real-world threats. Challenge D-2 focuses on incomplete realizations of threat models that could provide guarantees against the emergent threat posed by machine learning algorithms.

Challenge D-1: Protection against real-world adversaries and threat scenarios

There is currently a gap between security under a given (theoretical) adversary model and actual security requirements in a real-world scenario. Instead of trying to provide security guarantees under unrealistic assumptions such as the presence of a solely retrospective adversary, solutions should incorporate effective mechanisms to reduce the unauthorized use of published data during all stages of their life-cycle (such as preventing screen-capturing in Snapchat [192]).

The key challenge here is to develop adversarial models that represent real-world threats, that incorporate users' fears regarding their privacy and unintended exposure in real data publishing scenarios and to secure data sharing mechanisms under these models.

Challenge D-2: Protection against machine learning algorithms

Prevalent use of artificially intelligent systems by service providers adds a new threat dimension to the exposure of users' data. When the data is used to aggregate statistics or to train machine-learning models, e. g., for image classification or recommender systems, the information that data carries will implicitly remain in the model, even when the original data and everything explicitly linked to it is deleted. This limits users' control over the availability of information encoded in their previously shared data. Similarly, AI-based recognition algorithms also hinder users' capacity to effectively manage the visibility of their data from service providers. Despite some promising initial work, such as the use of adversarial examples [130, 145], it remains a challenge to counter the capabilities of AI systems and provide security guarantees against their use.

2.6. Further Issues

In Section 2.5, we presented a set of succinct, yet unresolved challenges regarding longitudinal online privacy management. Inherently, not all challenges can be approached from a purely technical perspective, e. g., challenges relating to flawed mental models require more holistic approaches, centered around end-users' issues. Our systematization is supposed to trigger activities in both the technical and the human-factor research communities, as a number of identified issues can only be resolved conjointly, taking into account both technical and user perspectives. One key takeaway is that technical solutions point towards promising directions, such as proposals targeting to overcome purely time-based exposure control mechanisms. However, it is critical to match users' actual needs in order to find adoption and to serve users by providing tools that they need to appropriately control the exposure of their personal online data.

We finally discuss five open issues that did not make it to our list of challenges because these were not directly derived out of the systematizations or were not specifically limited to publicly shared data. However, these aspects still provide further insights to the community about the landscape of longitudinal privacy of publicly shared data.

Control over Inversely Private Information Gurevich et al. [72] term an item of personal information about an individual *inversely private* if some party has access to it, but the individual does not. The situation described here elicits similar challenges as Data Co-ownership (cf. B-2) but is different in that users may not be aware of this particular information to exist. Daily interactions with various institutions ranging from toll roads operators to social networks generate vast amount of data about users. Processing users' private data and their pattern of interactions with the platform yields more inversely private data. In some cases, this private information held by companies can even con-

tradict users' preferences in the current context. For example, a social network user can continue to receive ads related to a preference derived from one of their old posts despite choosing to limit its lifetime. It is not straightforward to realize technical proposals that can allow users to manage and erase vast amounts of inversely private data about them held by different entities. The information is typically used for gaining a competitive edge, which is one of the reasons why corporations have been denying the inverse privacy entitlement to their users [72]. Regulations on service providers' processing of data could prove helpful, but it is unclear if existing laws, such as GDPR, provide users the right to erasure of inversely private information.

Content Obfuscation versus Usability While transformations targeting automated classifiers as means to solving the Security and Trust challenge (cf. D-2) may have only little impact on an image's appearance to humans, it also needs to be further investigated to what degree visible image perturbation is acceptable for users as a trade-off between privacy and vision comfort. There has been research on viewer satisfaction for blurring and pixelating photo scene elements that need to be protected [74, 105, 106], as well as on how the overall photo can be modified equally using aesthetic transforms to increase satisfaction [75].

Response to Privacy Paradox While users claim to be very concerned about their privacy, they nevertheless undertake very little to protect their personal data. Recent research on the privacy paradox has revealed discrepancies between users' preferences and their actual behavior [15, 44, 220]. Various studies have reported instances of users not taking the logical step of limiting the disclosure in their social networks despite being aware of privacy concerns [118, 136, 248]. These results hint that User Awareness (cf. C-1) alone is not going to lead to widespread adoption of longitudinal privacy technologies. To bridge the gap between users' desires and mechanisms' functionalities, it is equally important to

investigate and understand the causes and implications of the privacy paradox. Such an understanding will allow for design decisions that will increase the adoption of privacy-enhancing technologies.

Complications with Metadata Obfuscation Correlation and analysis of individual metadata can allow to draw conclusions about a person. Information deduced from communication flows can create privacy concerns in the same way as sensitive information obtained from posted contents [71]. Depending on the extent of metadata generation, sensitive information may still be preserved even if there is a technically perfect revocation mechanism for the actual data. For example, Facebook includes a feature that automatically adds descriptive keywords to photos to assist visually impaired users in comprehending its contents. In the case of photos of human subjects, their faces are detected, and users are suggested to enter the name of the person. While such features can be easily observed in the application interface, it remains unclear what types of additional data collection invisibly run in the background. One approach to counteract potential privacy threats by metadata aggregation and its residuals can be achieved by preventing metadata from being generated in the first place. This could be realized by applying image perturbation techniques to hamper metadata generation. While this strategy renders targeted classifiers unable to correctly assess image content, users would still be able to see the content. Related approaches have been developed with a different mindset, i. e., adversarial perturbations, e. g., used to interfere with traffic sign recognition used by self-driving cars [189]. More universal approaches to falsely classify images have also been demonstrated [130]. However, such protective mechanisms come along with new potential conflicts. Whenever the use of such a perturbation mechanism is transparent, or its presence becomes apparent, service providers (if considered in an adversarial setting) can adapt their classification techniques to circumvent the protection. This game-theoretic consideration,

already laid out by Oh et al. [145], is yet interesting to be investigated when developing even more sophisticated protection mechanisms.

Practicality of Referencing Data The current way to distribute data is to upload it to online platforms and copy-share it through various channels in order to make it available for different types of audiences [191]. In an entirely different approach, users could have only one instance of all their data hosted in a single location of their choice, providing them the individual level of privacy they desire. Instead of creating multiple copies of data and uploading them to different platforms, those services would be allowed or licensed to reference the data, without actually obtaining a copy or possessing them. Such a solution will enable tracking of all interactions with data objects and could facilitate the realization of challenging Expiration Conditions (cf. A-3). Bishop et al. [25] came up with ideas in a similar direction when discussing dissemination control as a means to manage online privacy.

The approach is not without challenges since interactions with the data entail modifications of the data itself. For example, multiple instant messaging platforms provide popular features enabling users to add text and drawings to the images sent in the chats. In such settings, each transformed output of the original data needs to be tracked in order to uphold the integrity of data provenance and ensure effective control over dissemination of the data.

In the light of applying such a scenario equally to end-users' personal data, one must also discuss if large companies such as Google or Facebook would already consider themselves such hosting platforms, providing almost every kind of service for one's online actions from a single source. It is unclear how the data object's single source of origin might impact its availability since providers would need to be willing to adapt their practices, and interfaces, to facilitate sharing of data hosted on their competitors' platforms.

2.7. Conclusion

We provided the first systematization to capture users' interactions related to longitudinal privacy management on existing platforms, as well as the landscape of diverse technical proposals dealing with the availability of online data. Our broad approach afforded us the ability to contrast end-users' desires and mental models against the technical proposals' use cases and adversarial assumptions. This enabled us to uncover open challenges and identify interesting problems where effective solutions have not yet been realized. By pointing the research community's direction towards these challenges, we hope that our work serves as an inspiration and a basis for the development of longitudinal privacy-enhancing solutions that will assist millions of end-users with managing the availability of their publicly-shared data.

3

User Perception of Message Deletion

Contents

3.1. Introduction

Over the past years, instant messaging applications on smartphones have become increasingly popular, with the most widely adopted messenger, WhatsApp, reaching approximately 2 billion monthly active users [204]. With more and more people using mobile messaging apps in their daily communication with their peers [196], it becomes harder for users to keep track of which information they share with whom. In addition to one-to-one conversations, messenger apps usually facilitate group chats and support various message types such as text, picture, video, or voice messages. Thus, remaining in control of one's own information exposure becomes an increasingly challenging task, also in the messaging domain. The increasing use of mobile messengers in everyday life carries the risk of accidentally sending messages to the wrong recipient. This can be a serious threat to users' privacy in general, especially when the communication contains sensitive personal information [84, 91].

3.1.1. Problem Statement

In mobile messengers, the course of a conversation is usually logged by each participant. Logging makes the communication persistent and may allow previously uninvolved third parties to retrieve past communication from the message history. Since communication in mobile messengers is often informal, it seems plausible that messages are often of ephemeral nature and not meant to be stored permanently [119]. However, users can freely decide to maintain their local message history, but also to delete specific messages from their own devices, e.g., to free memory on the device.

In October 2017, the messenger *WhatsApp* introduced a new feature which allows users to choose whether a sent message is to be deleted only locally or also from the recipient's conversation log [156, 208]. If users choose the latter, the message is replaced with a note indicating that the

message has been deleted. This also applies to messages the recipient has already read. The release of the *Deleting Messages for Everyone** feature indicates that the actual effect of the deletion functionality had not been explicitly stated before, thus raising the question whether the effects of such functions are apparent to the users. Other popular mobile instant messaging applications such as *Facebook Messenger* and *Skype* present the functionality for deleting messages in a similar fashion but have different effects.

This is interesting because, right now, users are bound to choices that designers and developers have made – long before when initially building their applications. It is unclear to what extent actual users and their feedback were involved in the underlying decision processes. In order to make devices such as smartphones better agents for their users, the capabilities of applications need to fit the users' needs. In particular, users should not have to face surprises because an effect triggered by their action does not match what they expected the action to do.

3.1.2. Contribution

To shed light on different realizations of message deletion in popular applications, we provide an overview of deletion functionalities in 17 applications. Based on this, we select three apps in which deletion has different effects despite being very similar in the way in which deletion is presented in the user interface. We use WhatsApp, Facebook Messenger, and Skype, to explore users' perception of message deletion. To this end, we we conduct a user study investigating (i) participants' preferences for the functionality of deletion mechanisms, and (ii) whether the participants understand the actual functionality of message deletion along with their consequences as implemented in instant messaging applications.

While our study explores users' preferences and expectations in messaging applications, it can also provide valuable insights for developers

*https://blog.whatsapp.com/10000635/Deleting-Messages-for-Everyone

to design features in their applications more comprehensible and usable. It has been well-known for almost two decades that failures in user interface design make it impossible for users to apply security features correctly [3, 101, 239].

Our major findings and contributions in this work are three-fold:

1. We show that those participants of our study who have deleted messages had various reasons for deleting messages, ranging from spelling correction to withdrawing messages that have been sent mistakenly or that are considered inappropriate in retrospect.
2. Regarding the scope of deletion, our results indicate that users appreciate to be able to select for each individual message whether it should only be deleted from their own device or also from the recipient's, as expressed by more than 40 % of participants in our study.
3. Our results indicate that the participants can better assess the effects of deleting messages when the functionalities are explained transparently. We reveal that the example implementation of WhatsApp can help developers to improve the user experience of their applications.

3.2. Deleting Messages

Mobile messaging, i.e., communication using mobile devices such as smartphones via apps such as WhatsApp, Facebook Messenger, or WeChat, has a large user base and is regularly used for personal communication with friends or family [113, 143]. Many of these apps offer the possibility to delete messages, while the concrete implementations widely differ between different apps.

We investigated the characteristics of the implemented deletion functionality for 17 popular messaging applications (cf. Table 3.1). We selected apps with a high number of monthly active users [197], concentrating on apps whose primary focus is messaging, and additionally in-

Table 3.1. Message deletion features in instant messengers.

Messenger		**Monthly Active Users [millions]**	**Local Deletion**	**Local Residuals**	**Global Deletion**	**Global Residuals**	**Separate Functions**	**Edit Message**	**Quote Message**	**Del. Received Msg.**	**Ephemeral Messages**	**Delete Conversation**
Facebook Messenger	1300*	[197]	●	○	○	–	–	○	○	●	○	●
GroupMe	11†	[198]	●	○	○	–	–	○	○	●	○	○
Hangouts	15†	[198]	○	–	○	–	–	○	○	○	◐	●
iMessage	—		●	○	○	–	–	○	○	●	○	●
Instagram Direct	375*	[41]	●	○	●	○	○	○	○	○	◐	●
KakaoTalk	50*	[199]	●	○	●	●	●	○	●	●	○	●
Kik	8†	[198]	●	○	○	–	–	○	○	●	○	●
Line	203*	[197]	●	○	●	●	●	○	●	●	○	●
Signal	—		●	○	○	–	–	○	●	●	●	●
Skype	300*	[197]	●	○	●	○	○	●	●	○	○	●
Snapchat	291*	[197]	●	●	●	●	○	○	○	○	●	●
Telegram	200*	[197]	●	○	●	○	●	○	●	●	●	●
Threema	—		●	○	○	–	–	○	●	●	○	●
Viber	260*	[197]	●	○	●	●	●	●	●	●	○	●
WeChat	1058*	[197]	●	○	●	●	●	○	○	●	○	●
WhatsApp	1500*	[197]	●	○	●	●	●	○	●	●	○	●
Wire	—		●	○	●	◐	●	●	●	●	○	●

*worldwide, †US-only, — no information available;
● provides functionality, ◐ partially provides functionality, ○ does not provide
functionality, – functionality does not apply;

cluding messengers focused on protecting user privacy, such as Signal or Threema.

Our overview captures the situation as of January 2019, whereas application developers may adapt the functionality for subsequent versions. All properties were examined in a scenario where two individuals engage in a one-to-one conversation, both using a single mobile device such as a smartphone. We identified conceptual differences between the implementations, which we discuss in the remainder of this section.

3.2.1. Local vs. Global Deletion

The effects of deleting a message differ between applications. Except for Google Hangouts, all applications under consideration (cf. Table 3.1) support *Local Deletion* from the conversation history of the sender's device. Hangouts differs in that it only allows to delete the entire conversation history with the respective contact.

The majority of applications also allow messages to be removed from the recipient's device, denoted *Global Deletion*. Popular applications supporting this feature include WhatsApp, WeChat, and Skype.

We say a messenger provides *Separate Functions* if it allows the user to determine the scope of deletion. This property only applies to messengers supporting both types of deleting messages, i. e., locally and globally. If a message can only be removed from all devices at the same time (which applies to Instagram Direct, Skype, and Snapchat), this messenger does not provide separate functions. Among those messengers providing separate functions, we observe two flavors of separation. WhatsApp, KakaoTalk, and Wire let the user choose between the two options *Delete for me* and *Delete for everyone* in a prompt appearing after the message has been selected to delete. WeChat, Line, and Viber provide two distinct menu items for these functions.

There are several messengers explaining the effects in a dialog that has to be confirmed. For example, Facebook explains that the user can only

"delete [their] copy of the message". In KakaoTalk, where the user can explicitly choose between *"Delete for me"* and *"Delete for everyone"*, selecting local deletion triggers a reminder that *"the message will only be deleted from your chatroom and will still be visible to your friend(s)"*. In contrast, Skype only provides one *"Delete"* functionality that deletes the message for all participants in the conversation, without providing any further explanation or choice. Given these different levels of detail in explaining functionality, we think that the effects of the deleting mechanism of a particular messenger are not always obvious.

When a message is deleted globally, several messengers also confirm this in a dialog, along with possible limitations of the functionality. Line and KakaoTalk explain that deleting for everyone may not work depending on the application version used by the other participants in the conversation. WeChat and WhatsApp also show this hint, but only when the functionality is used for the first time.

The naming of local and global deletion can also be used to make users aware of their different functionality. While local-only deletion is mostly called *"Delete"* (except for *"Hide message"* in GroupMe), there are various names for deleting a message globally. In Instagram Direct and Line this feature is called *"Unsend"*, WeChat names its global deletion *"Recall"*. In some cases, deleting a message globally is only available within a specific time span, ranging from two minutes in WeChat to 24 hours in Line.

If a user can identify when messages in a conversation have been deleted, we say the messenger leaves *Residuals*. The applications providing local-only deletion never show any residuals when a message is deleted locally. When a message is deleted globally, KakaoTalk, Line, Snapchat, WeChat, and WhatsApp leave a hint in place of the former message within the conversation, stating that a message has been deleted. The hint appears on the devices of all participants in the conversation. Wire follows a different approach: On the recipient's device, it replaces the message with only the name of the sender; it does not leave any residuals on the sender's side.

3.2.2. Deleting Quoted Messages

Several messaging applications include features to reply to a message, i. e., to send a new message in which the original one is embedded as a quotation. We examined how the messengers offering a reply functionality handle the interplay of replies and deleted messages in three scenarios.

(1) Alice sends a message to Bob, Bob replies to that message, and then Alice locally deletes the original message from her device.
(2) Alice sends a message to Bob, Alices deletes it locally, Bob then replies to the message.
(3) Alice sends a message to Bob, Bob replies to that message, and then Alice globally deletes the message. This case only applies to applications offering global deletion.

Our observations are listed in Table 3.2. Only in Line and Wire the quoted message is deleted along with the original message in all three scenarios. Instead of the original message, both applications embed a notification stating that the message is not available. In Telegram, the quoted message is only deleted in the second scenario, i. e., when the original message is deleted before the recipient has replied. In this case, Telegram only shows the reply as a standalone message. In all other messengers, deleting does not affect quotations of a message. In the second scenario, Signal shows an embedded notification along with the quote, stating that the original message could not be found, while still displaying the original message.

3.2.3. Additional Properties

Three messengers, Skype, Viber, and Wire, allow users to edit the content of a message in retrospect. In Skype and Viber, we were able to edit messages two days after they had been sent, but we could not determine if these messengers have a time limit for editing. All three applications indicate if a particular message has been edited.

Table 3.2. Deleting quoted messages.

Messenger	Send, Reply, Delete Local	Send, Delete Local, Reply	Send, Reply, Delete Global	Notification
KakaoTalk	●	●	●	none
Line	○	○	○	*"Message unavailable"*
Signal	●	●	–	*"Original message not found"*
Skype	–	–	●	none
Telegram	●	○	●	none
Viber	●	●	●	none
WhatsApp	●	●	●	none
Wire	○	○	○	*"You cannot see this message"*

● original message visible; ○ original message not visible; – scenario does not apply

The majority of messengers we considered allow users to delete received messages from their device. This functionality applies to all messengers except for Hangouts, which does not allow deleting individual messages at all, and the messengers that do not allow local-only deletion, i. e., Instagram Direct, Skype, and Snapchat. Deleting a received message only takes effect locally, i. e., the recipient cannot delete the message from the sender's device.

Five messengers we considered support some concept of *Ephemeral Messages* but differ in their implementations. An ephemeral message is automatically deleted from all devices in the conversation after a specific time span. Instagram Direct provides ephemerality on a per-message-basis and limits it to media content such as photos, which automatically disappear 10 seconds after being displayed. In other applications, ephemerality is configured on a per-conversation-basis. Both participants can change the conversation settings that always affect both sides. Signal and Telegram allow to set time spans between a few seconds and one week. The time span can be changed during the course of the conversation, but the new time span only applies to future messages and does not affect older messages. Similarly, Snapchat users can set a timer for the conversation but only have two options, immediately after viewing

and after 24 hours. Hangouts allows users to turn off the conversation history, but the message expiration time is neither explicitly specified nor configurable. Contrary to the other applications, ephemeral messages in Hangouts do not expire individually but in groups.

The functionality to delete an entire conversation is an additional feature supported by all mobile instant messaging applications we considered, except for GroupMe. Deleting a conversation only takes effect locally and can basically also be achieved by manually deleting all messages in the conversation. However, in Hangouts, deleting a conversation is the only way to delete messages locally.

3.2.4. Study Goals

The different messaging applications comprise a variety of implementations of deletion functionalities. We consider this a broad selection of offers made by the application developers to their users. From the opposite perspective, this directly raises the question which features users actually prefer for their everyday conversations.

Therefore, we first study how commonly users delete messages, whether there is a need for this functionality, and in particular, which options users prefer, inspired by the currently available options. Additionally, we examine whether users can correctly assess the capabilities of deletion functions and whether we can identify differences in distinct implementations of these functions. We expect that the variety of implementations of deletion mechanisms is confusing for users.

For example, deleting a message in Skype removes messages from all participants' conversations, whereas the identically named function in, e. g., Facebook Messenger only removes messages from the sender's log. Line Messenger is more transparent by providing a prompt stating that the message is only deleted locally, and requiring the user to confirm before the message is eventually deleted. It is unclear how an average user who has not explicitly explored the actual impact of deleting in a

particular app can objectively assess what happens when a message is deleted.

3.3. Method

In this section, we discuss the design and method of our study in detail. We conducted a between-subject study comprising three test conditions (study groups). During the study, the participants in each condition interact with one particular mobile instant messenger and answer 16 questions on a laptop.

3.3.1. Test Conditions

For our practical study, we assigned participants one of three test conditions based on the different instant messaging applications:

- **Skype** (version 8.13) deletes messages from the message logs of all participants in the conversation;
- **Facebook Messenger** (version 151.0) allows the sender to delete messages from their own conversation history only;
- **WhatsApp** (version 2.18) allows users to select whether to delete the message just from the sender's conversation history or for all parties involved.

These messengers were selected because they have a large user base and they implement different behaviors of the message deletion functionality, thus representing all sound variations of deleting messages (cf. Table 3.1). During the assignment, we did not take into account if a participant had used the respective messenger application before. All versions correspond to the most current versions available as of February 2018.

3.3.2. Study Design

Our study comprised five steps: (1) Introduction, (2) Practical Task: Writing and Deleting a Message, (3) Questionnaire Part I: Preferences for Deleting Messages, (4) Questionnaire Part II: Reconsidering Effects of Deleting, and (5) Debriefing.

After the introduction, the participants completed an initial practical task in which they sent and deleted a message using an instant messaging application. Subsequently, participants answered a two-part questionnaire, during which the interviewers showed them the result of the experiment. We explain these steps in more detail in the following.

Step 1 – Introduction

In the first step, we explained each participant the reason for and purpose of the study as well as the study procedure. Furthermore, we informed them that we did not collect personal data, how long participation typically took, and how we compensated them.

Step 2 – Practical Task: Writing and Deleting a Message

Next, the first stage of the practical task followed. We asked the participants to write, send, and delete a message using a specific instant messaging service.

For this task, we gave the participants a mobile phone (*Samsung Galaxy S6* running *Android 7*) with the specific messaging service already opened to keep the task of sending and deleting a message as simple as possible – we did not ask the participants to use their own mobile devices. We used our lab mobile phone in order to create a more controlled environment where the messaging service was installed and working, and the contact phone number was already in the contact list. Participants were asked to type an arbitrary message, but if they struggled to come up with a message of their own, we suggested them to send “hello”.

On a second phone, we then showed the participants that the message had arrived at the recipient's device, and asked them to delete the message on the device they had used to send the message. If necessary, we assisted the participants to figure out how to delete the message. At this point, we did not show them the effect of deleting the message on the recipient's device – what we deferred until Step 4.

Step 3 – Questionnaire Part I: Expectations of Deleting Messages

At this point, the participants started to fill out a questionnaire. The first part of the questionnaire consisted of a few warm-up questions about the participants' usage of mobile devices and instant messaging, and whether they deleted instant messages and why. It further contained questions about the participants' expectations concerning the experiment – whether the message was deleted everywhere or only from the sending device. Additionally, we asked the participants which deletion behavior they would prefer. Demographic data was also collected in this part of the questionnaire.

Step 4 – Questionnaire Part II: Reconsidering Effects of Deleting

Before the participants proceeded with the second part of the questionnaire, we revealed the outcome of the experiment by showing the participants the message history of the recipient's device. This allowed them to see the effect of deleting the message on the recipient's side.

Subsequently, the participants continued with the second part of the questionnaire, specifically focusing on questions about the message deletion and whether it behaved as expected. In the last two questions, we asked the participants whether there should be limitations for deleting messages from the recipient's message history.

These questions were primarily addressed to participants of the WhatsApp and Skype conditions since these messengers allow deleting messages from the recipient's message history.

Step 5 – Debriefing

After these final questions, we thanked the participants for their participation. When they had any questions about the study, we answered them in this step.

Finally, we deleted the entire message history to preserve the privacy of the participants and to allow the next participant to start with an empty message history.

3.3.3. Pilot Study

In December 2017, we conducted a pilot study to evaluate the procedure of the study, determine the duration per participant, and test the comprehensibility of the questions. We tested the study on 8 colleagues from a co-located department (75 % male, 25 % female, age ranging from 25 to 59 years). The participants did not have any prior knowledge of the study and its goals. As a result, three questions were removed from the questionnaire as they turned out to be somewhat redundant or too imprecise. We also decided to let participants fill out the questionnaire on laptops instead of structured interviews or paper-based questionnaires to avoid errors during data collection and simplify administering the responses.

3.3.4. Study Protocol, Recruitment, and Demographics

Over a period of three days in February 2018, we collected a total of 135 responses from visitors to the main cafeteria of Ruhr University Bochum, located in the largest metropolitan area in Germany. The main cafeteria is centrally located and frequented by students and staff from all depart-

ments. We set up two tables in relatively quiet corners near the two main entrance doors and recruited participants from the passing students and staff. This setup allowed for rather quick recruitment of participants but may also have biased the sample. However, as the cafeteria serves all departments, we expected participants with a wide variety of backgrounds.

Study participants could choose if they participated in the study in English or German. 93 % of the participants chose to answer in German. Completing the study took on average five minutes, and we compensated each participant with two chocolate bars regardless of whether they completed the study or aborted early.

Although all participants completed the study, we discarded 10 responses because of incomplete answers, resulting in 125 responses we used in the evaluation. The participants were randomly assigned one of the three conditions.

When the participants answered the questionnaire, the interviewers kept their distance in order to not create additional pressure, while staying available for questions. We did not record the exact number of questions raised by participants, but we estimate that less than five participants asked for clarification of survey questions.

We collected demographic data from the study participants. 32 % of the participants stated to be female and 64 % identified as male. The median age is 25 years in a range from 18 to 75 years. Table 3.3 summarizes the response to the demographic questions.

We also asked the participants to self-estimate their proficiency in using mobile devices on a five-point scale from beginner (1) to expert (5). According to their answers, more than 60 % of the participants rated their experience in using mobile devices as four or five.

3.3.5. Response Preparation

For three questions (i. e., Q5, Q14, and Q16), our participants were asked to provide their responses as free text. We used a coding approach to

Table 3.3. Participant demographics.

	Facebook	**Skype**	**WhatsApp**	**All Conditions**	
Age					
<20	9	10	6	25	(20.0 %)
20–34	30	26	32	88	(70.4 %)
35–49	4	3	1	8	(6.4 %)
≥50	1	2	0	3	(2.4 %)
No answer	1	0	0	1	(0.8 %)
Gender					
Female	11	12	17	40	(32.0 %)
Male	31	28	21	80	(64.0 %)
Other	2	1	0	3	(2.4 %)
No answer	1	0	1	2	(1.6 %)
Level of experience with mobile devices (self-assessed)					
1 (beginner)	0	1	0	1	(0.8 %)
2	5	3	3	11	(8.8 %)
3	12	11	7	30	(24.0 %)
4	19	15	16	50	(40.0 %)
5 (expert)	7	10	11	28	(22.4 %)
No answer	2	1	2	5	(4.0 %)
Total	45	41	39	125	(100.0 %)

prepare the responses for analysis. Three authors independently created codes based on the free text responses and assigned each response one or more codes resulting in three independent codings per question.

Subsequently, one of the coders created a code book for each question based on the three individual codings. The code book comprised a list of keywords, each accompanied by a short descriptive sentence. Creating the code book required minor modifications such as renaming or merging particular keywords.

All changes in the individual codings have been documented and required approval of the respective coder. When the authors had used different codes for a response, this response was assigned the union of codes assigned by the three coders. In the last step, the code book was approved by the three coders.

3.3.6. Ethical Considerations

Our university does not have an IRB or ethics board which covers the type of our study. However, we have taken great care to adhere to principles of ethical research [100]. Our study was designed such that it did not contain deceiving questions. In case participants asked immediately after deleting the message how the deletion affected the recipient's message history, we asked them to be patient until they had completed the first part of the questionnaire. Furthermore, we did not store any data which would allow us to link participants to their responses.

In the recruitment process, each participant was informed that they were participating in a scientific study, about the purpose of the study, the possibility to withdraw at any time without giving any reasons, and that no personally identifying information would be stored.

At the beginning of the questionnaire, the participants were shown an introductory text summarizing the information previously given orally during recruitment.

We also informed the participants about the estimated duration of the study (approx. five minutes) and their compensation (two chocolate bars). However, some participants required up to 10 minutes or more because they provided detailed answers to the free text questions. Answering these questions was not mandatory and could be omitted. The demographic questions were also completely optional, and the participants could skip them without providing an answer or choose the *I prefer not to answer* option.

3.4. Results

In this section, we present the results of our study. We report our findings on the participants' preferences and expectations of deleting messages. The results from the practical task to delete a message are presented

and analyzed as to whether and to what extent users correctly assess the actual capabilities of deletion functionality.

3.4.1. User Preferences for Message Deletion

First, we consider the participants' preferences for the functionality of deletion mechanisms as expressed in the questionnaire. Here we are faced with subjective wishes and concerns of users. First, we analyze the prevalence of deleting messages, i. e., if users actually use message deletion features in their daily lives, how often, and with what intentions they use them. Subsequently, we analyze users' preferences regarding several features of deletion to find out what technical implementation they think best fits their needs.

Frequency of Message Deletion

To learn about the prevalence of message deletion, we directly asked the participants how often they delete messages in instant messaging (*Q4: How often do you delete instant messages?*). Response options ranged from "Several times a day" to "Almost never", including "I don't know". The distribution of responses is shown in Figure 3.1.

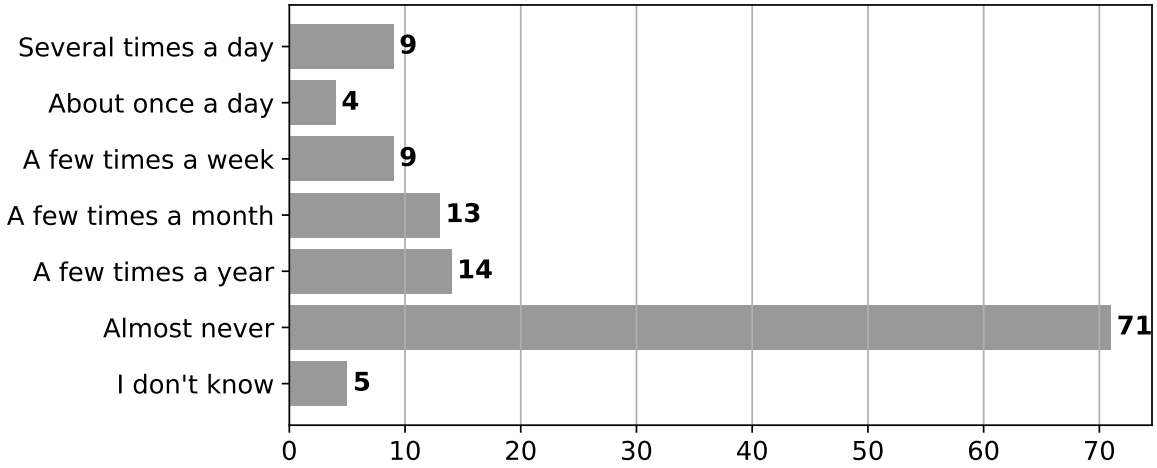

Figure 3.1. Frequency of responses to *Q4* (How often do you delete instant messages?).

On the one hand, we can see that, on average, message deletion is a relatively infrequent event: 56.8 % of users ($k = 71$) almost never delete messages, and only 10.4 % ($k = 13$) of participants use it on a daily basis. Among those 39.2 % ($k = 49$) of participants who have used message deletion before, we see that usage patterns widely differ. We find about equal numbers of participants using message deletion "a few times a year/month/week" as well as "several times a day" (each approx. 10 %).

Thus, the results from our sample of participants do not show a clear trend regarding the prevalence of message deletion. They also do not capture differences between actual message deletion and mere editing, which was provided as a reason for deletion by 13.6 % ($k = 17$) of our participants.

Reasons to Delete Messages

Participants could describe their reasons for deleting messages in a free text response, since we expected a wide variety of answers.

We derived 11 codes from our participants' responses which we assigned 68 times to 42 responses, following the coding approach described in Section 3.3.5. Three responses were left out as the coders agreed that they were too ambiguous. The frequency of each code is shown in Figure 3.2.

Revising messages was the most frequently named reason to delete messages (*"Usually, I don't [delete messages], except for typos"*† (P115)). We coded these responses as *revision* ($k = 17$) since they indicate that the participants delete messages with the intention of replacing or editing them instead of removing them. Participants stated that they revise messages *"because they contain mistakes (typos)"* (P33) or to *"reconsider the wording"* (P34). Conversation consistency may also play a role when deleting messages: *"If I misspelled something and nobody has read it yet"* (P94).

†In the following, German-language quotes by participants were translated to English by the authors.

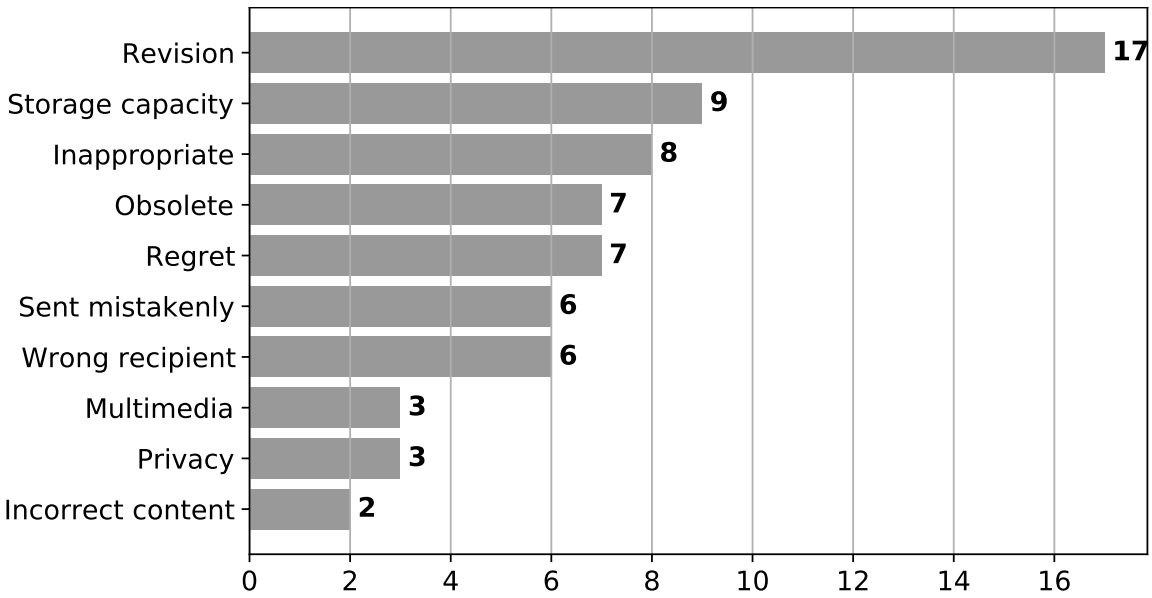

Figure 3.2. Frequency of codes for responses to *Q5* (What are your reasons for deleting messages?). We collected a total of 42 responses to which we assigned the codes 68 times.

Participants also stated that they delete messages *"if [...] inappropriate"* (P129) and *"sometimes [...] because I have said something inappropriate"* (P74). They consider some of their messages as *inappropriate* ($k = 8$) in retrospect and delete them for this reason.

Most of the responses coded as *inappropriate* indicate some sort of *regret* ($k = 7$) of having sent the message in the first place. Explanations reported by participants include that they *"texted without thinking"* (P122) or because of *"spontaneous emotions that [they] regretted afterwards"* (P36).

Messages are also deleted when they are considered *obsolete* ($k = 7$), for example if they are *"no longer relevant"* (P76) or *"the circumstances under which I had sent the message have changed [...]"* (P52).

Participants further explained that they deleted messages they had *sent mistakenly* ($k = 6$): *"For example, because I have sent something twice"* (P80)), *"wrong unintended messages"* (P12). Another closely related reason to delete messages is if messages have been sent to the *wrong recipient* ($k = 6$). Participants described that they deleted messages because they had sent them to the *"wrong recipient or the message was*

stupid and [they] wanted to take it back" (P40). Other similar responses were: *"Sent to the wrong group/person or just because it was not clear enough"* (P2) and *"typo / send to wrong person"* (P110).

Another frequently mentioned motivation for deleting messages is to free memory on the mobile device. Nine of the participants gave responses including *"lack of memory"* (P39), *"no memory"* (P53), or *"just because they consume some memory"* (P84) that suggest limited *storage capacity* ($k = 9$) as a reason to delete.

Text messages do not consume much memory, but *multimedia* content such as images or videos does. We assume that the participants mentioning *storage capacity* issues also had *multimedia* content in mind. However, only three responses explicitly referred to *multimedia* content, e. g., P25, who wrote *"media files that consume too much memory or group chats that are not interesting"*.

In summary, we can distinguish three major categories: Users delete messages to make corrections, free storage, or for privacy reasons. We consider the following reasons for deletion as privacy-related: messages being *inappropriate*, *obsolete*, *regretted* by the sender, *sent mistakenly*, or to the *wrong recipient*. Thus, 54.4 % ($k = 37$) of the codes are somehow privacy-related, distributed across 61.9 % ($k = 26$) of the ($n = 42$) free text answers we have received, that is 20.8 % of all 125 participants.

Preferences for Deleting Messages

We have asked users about their preferred variant for deleting messages (*Q7: Which of the following do you prefer when you delete a message?*), i. e., from which message histories they prefer messages to be removed. Participants could pick one of four predefined answers. The results are listed in Table 3.4. The majority of participants (84 %, $k = 105$) preferred either the message to be deleted from both the sender's and recipient's logs or to be given the choice between global and local deletion whenever they delete a message. These numbers are supported by our ob-

servations of the study participants who were assigned to the WhatsApp condition in the experiment. 36 of them chose the *Delete for Everyone* option, while only three decided to remove the message from their message history only. This indicates two things: First, our results suggest that the majority of users who have decided to delete a message prefer deletion to have global effects. Second, there also appears to be a need for a selection mechanism on a per-message-basis, which implies that users desire more granular functionality and also higher transparency when they delete messages.

Table 3.4. Preferences for deleting messages (responses to *Q7*).

Preferred Option	**Responses**
The message is deleted from my device only.	12 (9.6 %)
The message is deleted from the recipient's device only.	8 (6.4 %)
The message is deleted from both devices.	54 (43.2 %)
For each message, I can choose where to delete from.	51 (40.8 %)

Preferences for Notifications

We further asked participants how they perceived notifications in the contexts of messaging and deletion. First, we asked users about notifications whether a message has been read (*Q8: Do you want to be notified if the recipient has already read the message?*). The majority of participants (77.6 %, $k = 97$) preferred messengers to provide such notifications. Second, we asked about residuals in place of a deleted message (*Q9: Do you think that the recipient should be told that the message has been deleted (e. g., through a "message deleted" hint)?*). 63.2 % of participants ($k = 79$) stated that this type of notification should not appear in the conversation.

User Preferences for the Limitations of Deleting Messages

We have asked users whether the (global) delete functionality in instant messaging should be limited (*Q15: Do you think the delete function should be limited?*). We suggested examples such as time limitation, message order, or message status (read vs. unread). While 39 participants (31.2 %) agreed with this, we received 86 negative answers.

The 39 participants who supported such a limitation were asked to further specify the type of limitation (*Q16: How should the delete function be limited?*). We coded their free text answers into six different categories. While we did not categorize seven answers as we agreed that these were too ambiguous or not related to the question, we assigned a total of 35 tags to 32 different answers. The distribution of the answers is illustrated in Figure 3.3.

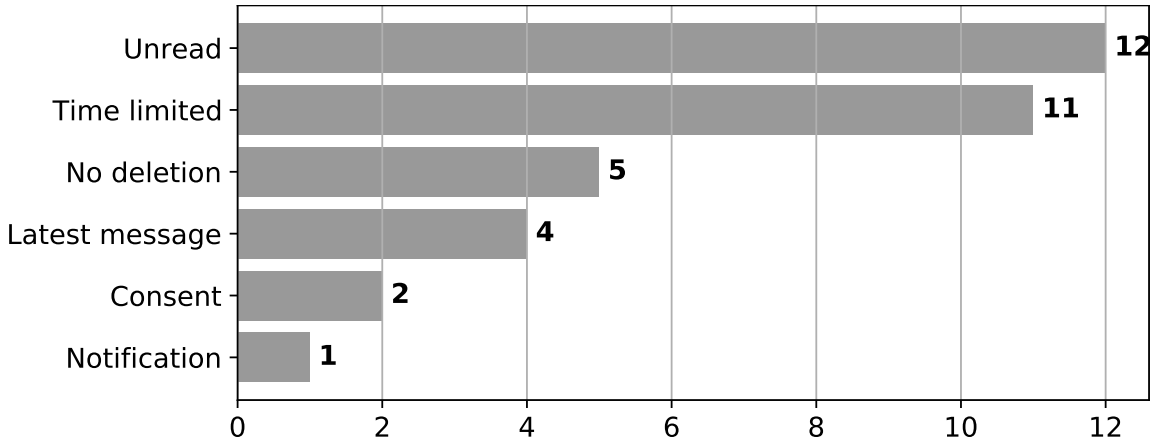

Figure 3.3. Frequency of codes for responses to *Q16* (How should the deletion function be limited?). We collected a total of 32 responses to which we assigned the codes 35 times.

The most frequent proposals were to either allow deletion only for *unread* messages ($k = 12$) or to limit the deletion functionality based on *time* ($k = 11$).

Arguments in favor of restricting deletion to unread messages include possible manipulation (*"because they have not yet caused a reaction on the recipient's side. If any message can be deleted, the recipient can probably be led to believe they had only imagined the deleted message to*

exist, which could be exploited [...]" (P69)) and discomfort with conversations partially disappearing from the recipient's conversation history (*"It makes me feel uncomfortable if I cannot look up conversations that have already taken place"* (P25)).

For time-restricted deletion, the suggestions for how long the functionality should be available range from *"5 minutes"* (P89) to *"24 hours"*, with one hour being the most common proposal, suggested four times. Participants reported to favor time-restricted deletion because of *"typing errors or [if] the message was supposed to be sent to another recipient"* (P39) as opposed to *"things [...] being distorted in the long run"* (P56) and the need of a consistent message history to *"prove things"* (P67). These answers suggest that participants see a connection between the time and purpose of deletion. Another participant argued that the recipient could be expected to have read the message after a certain time has passed, so being able to delete it later would *"no longer [be] worth it"* (P43).

Five participants expressed that they opposed message deletion in general as *"some information could be important for another person"* (P128) or because *"it creates an illusion of deletability that cannot be satisfied – just think of screenshots"* (P101). Discomfort with others manipulating information already stored on one's device was also mentioned (*"It shouldn't be possible to delete data you know to exist on your device. Especially if you are not notified of it."* (P112)).

Four participants proposed to restrict deletion to the *latest message* only. This restriction is actually implemented in the WeChat messenger (which we did not cover in our study) and appears interesting in that it keeps the conversation history consistent. Deleting a message after one or more follow-up messages can change the entire context of the subsequent conversation.

Quite interestingly, two participants suggested that for each conversation all partners should be required to *consent* whether and under which circumstances messages can be removed from the conversation history.

People should be able to *"pick the messenger that best matches [their] needs"* (P112), including the need for a deletion functionality. *"Before the conversation begins, every participant should be able to determine if [...] and for how long message should be able to be deleted. [...] If the circumstances permit or require it, a new conversation could be generated (as in 3-person settings on Facebook)"* (P112).

3.4.2. User Perception of Message Deletion

Next, we analyze users' perception of message deletion, i. e., what our participants expect to happen when they delete a message, and whether the actual outcome of deleting a message matches what they expect.

Expectation Matching in Real Implementations

We asked our participants about their opinion from which devices their message has been deleted (*Q6: We just asked you to send a message and then to delete it. What do you think – where has the message been deleted?*). Participants could choose each of the two devices involved in the conversation and specify additional answers as free text. In Facebook Messenger, the participants who only selected the sender estimated the outcome of the deletion correctly. In Skype, the outcome is correctly estimated when participants selected both the sender and the recipient. In WhatsApp, there are two options, sender only and both parties, depending on what participants actually selected when they deleted the message in the experiment – we consider both cases a correct prediction. We did not capture if participants had experience with the messenger they tested or if they knew about its actual functionality before.

In Step 4 of the study, we disclosed to the participants whether the message they had sent and deleted was still available on the recipient's device. We then asked the participants if the result matched their expectations (*Q13: Does this result match your expectations?*). Additionally, the participants could provide a free text answer to specify differences be-

Table 3.5. Response frequencies for Q6, Q13, and Q14.

	Facebook	Skype	WhatsApp
Q6: Where has the message been deleted?			
Sender Only	38	26	8
Recipient Only	0	3	5
Both	7	9	26
Correct Answers	38	9	34
Q13: Does this result match your expectations?			
Yes	32	20	31
No	13	21	8
Q14: Why does this result not match your expectations?			
No Deletion	10	0	0
No Deletion Note	1	1	0
No Msg Read Note	0	1	0
Deletion	0	8	2
Deletion Note	0	3	5

tween their expectations and the result (*Q14: Why does this result match your expectations? Why not?*). The responses to these three questions (i. e., Q6, Q13, and Q14) are summarized in Table 3.5.

Overall, 66.4 % of the participants ($k = 83$) stated that the observed behavior matched their expectation; however, the results depend on which messenger was used. For Facebook Messenger 71 % agreed, for Skype 49 % agreed, and for WhatsApp 80 % agreed.

We used a chi-square test of independence to test if these differences among the three messengers are statistically significant and found a significant influence ($\chi^2 = 9.1468$, $df = 2$, $p = 0.01032$). For post-hoc testing we used chi-square tests on pairs of messengers and applied corrections for multiple testing. We used Bonferroni correction as a conservative choice, as the number of tests is small. Among the post-hoc tests on pairs of messengers, we found significant differences between Skype and WhatsApp ($\chi^2 = 6.8806$, $df = 1$, $p = 0.02613$). Participants in the WhatsApp condition could better assess the effects of message deletion

by 30 % than participants in the Skype condition. A summary of the test results is shown in Table 3.6. We used a significance level of $\alpha = 0.05$.

Table 3.6. Indepence test results.

Messenger Combination	*df*	χ^2	*p*
Facebook vs. Skype	1	3.58980	0.17439
Facebook vs. WhatsApp	1	0.39886	1.00000
Skype vs. WhatsApp	1	6.88060	0.02613
Omnibus (All Messengers)	2	9.14680	0.01032

Evaluating the results for Q14, we realized that Q13 can refer to multiple dimensions of message deletion instead of just the question whether the message was deleted from the recipient's device or not. Some participants stated that they had correctly anticipated the message being deleted (or not) but were surprised by other aspects of the process such as deletion notifications or (the lack of) other residuals on the recipient's device. Consequently, they provided a "no" answer even though they had correctly predicted which devices the message would be deleted from. This makes it harder to assess the binary answers to Q13; in retrospect, a more fine-grained answer space should have been provided. We address this issue in the Limitations section.

Reasons for Non-matching Expectations

Prior to the experiment, we expected a higher rate of expectation matching, particularly in the WhatsApp condition, where participants were able to explicitly choose which message history they would like to delete the message from. Therefore, we analyze the reasons why the expectations did not match. Participants could specify in detail the reasons why and how the result differed from what they had expected (*Q14: Why does this result match your expectations? Why not?*).

We received 32 free text answers and coded them, again, as described in Section 3.3.5. One participant noted to have expected a prompt to

choose whether the message should be deleted locally or globally. The coders agreed to drop this answer as it is not related to the disclosure of the result at the end of the experiment. We categorized the remaining 31 answers into five disjoint categories as illustrated in Table 3.5.

The majority of responses ($k = 20$) simply referred to surprises because a message was deleted (*Deletion*) or because it was not deleted (*No Deletion*). Messages deleted from the recipient's device surprised participants who did not expect global deletion to work at all, only with unread messages (*"I thought if a message has already been read, it can no longer be deleted from the recipient's device"* (P40)), or only with certain messengers (*"I though it only worked in WhatsApp"* (P75)). Participants also expressed concerns about the concept of global deletion in general, including *"possible conflicts [and] evidence [being] deleted permanently"* (P57) or stated that *"once data has been transferred, the recipient should be able to manage it on their device autonomously"* (P133). Conversely, eight participants in the Facebook Messenger condition stated they had expected the message to also disappear from the recipient's device, with one explicitly referring to their experience with other messengers (*"I use Telegram, so I'm used to being able to delete messages from both devices [...]"* (P94)), and another questioning the concept of local deletion because *"it contradicts the purpose of deletion if the recipient can still read the message"* (P73). Another participant in this condition thought technical problems *"such as a failed connection to the server or (probably intentional) client malfunction"* (P97) were to blame for the message still existing on the recipient's device.

Ten participants referred to the delete notification as the reason why the outcome did not match their expectations. Two of them had expected the recipient to *"at least"* (P8) be notified that a message has been deleted *"because [...] this happens from experience, e. g., on WhatsApp"* (P105). In turn, eight had expected the message to disappear without a trace and were surprised by the delete notification because it *"sparks mistrust"* (P36) or *"doesn't matter [...] and it doesn't convey any message"* (P48).

The reasons for mismatched expectations do not apply to all three messengers equally, e. g., only participants who used Facebook Messenger could expect a deletion that did not occur ($k = 10$). Quite interestingly, the answers indicate that the expectation mismatch partially originated from the notification that a message has been deleted (Skype: 3, WhatsApp: 5).

3.4.3. Limitations

We have planned and conducted our study thoroughly. However, our sampling approach introduces certain limitations. We have reached a large number of participants with moderate effort, but this resulted in a sample biased towards younger people who have (at their own judgment) higher than average experience with mobile devices. For better general applicability of our results, a sample with a more representative age distribution and more objective assessment of experience would be desirable.

The study environment for the practical task and answering the questionnaire was rather busy compared to an in-lab setup, which is, however, more representative for normal smartphone usage.

In our survey, several questions only offered binary (yes / no) answer options. Most of the binary answers were used in the warm-up questions. Only the answers to Q13 were used for quantitative evaluation, and these are supported by the qualitative answers to Q14. Answer ranges based, e. g., on Likert scales might have been a better instrument to capture varying levels of people's opinions. Our goal was to obtain a coarse estimation of expectations on message deletion, not necessarily representing all possible aspects. The use of a survey with predominantly closed questions facilitated the analysis compared to interviews, at the expense of limiting the participants' ability to express differentiated answers.

The test conditions were also limiting the applicability of our findings, in that we only tested three different implementations of messengers and

did not cover all deletion features such as ephemeral messages. The three messengers we tested are, however, among the most popular ones and comprise different realizations of the deletion functionality.

3.5. Discussion

The term *message deletion* can be ambiguous as it can be unclear whether messages are removed from the sender's or the recipient's log, or both. Users could not always correctly estimate the consequences of a particular deletion action. Participants in our study could not correctly assess the actual effects of deleting a message in an application that does not adequately explain its functionality. This mismatch could possibly be remedied by improving the interface design, i. e., better explaining the consequences of selecting *delete*. A convenient example for this is WhatsApp's implementation which lets users directly decide whether they prefer a message to be deleted only locally or also on the recipient's side. While WhatsApp's implementation is the most transparent one, it also meets the desire expressed by a considerable number of participants. Overall, 84 % preferred to be able to delete messages from the recipient's device, either by always deleting on both devices or by having an explicit choice between local and global deletion.

We have seen that our participants consider message deletion a useful feature they would use in a diverse range of ways. Since 13.6 % of our participants indicated that they deleted messages to revise them and send them again instead of actually removing them, we recommend application developers to consider including a dedicated *edit* feature into their applications.

It is still to be investigated whether a more clear description of the delete function on the user interface can better clarify where messages are deleted, even when no choice is given to the user. One example could be the Line messenger, which explicitly advises a user that the respective

message is only deleted from the user's local conversation history and that the recipients will still be able to read it.

It is interesting that a majority of participants (68.8 %) did not express an explicit desire for limits on the delete functionality. Participants in favor of such limitations explained that they desired preserving a consistent conversation. The limitation to seven minutes originally implemented by WhatsApp appears appropriate according to the majority of reasons users stated for deleting messages. This time span is sufficient to correct or improve messages and to withdraw messages that have been sent mistakenly or to a wrong recipient. However, it remains unclear how this limitation was determined. In early March 2018, the time limit for message deletion in WhatsApp was extended to 68 minutes and 16 seconds (i. e., 2^{12} seconds) [134], which suggests that the rationale for the concrete time limit may also be purely technical.

Another interesting proposal – yet not implemented in any of the messenger applications we have examined – might be consent-based deleting. In such a scenario, messages can only be deleted if all participants in the conversation have explicitly stated so beforehand, on a per-conversation basis. Such a mechanism could balance individual interests of both the sender (to keep control of potentially sensitive data) and the receiver (to keep track of the conversation). Unlimited availability of the functionality to delete messages could evoke malicious deletion, e. g., to alter the context of a conversation retroactively. Consent-based deletion might help to reduce these threats.

These examples show how the user experience of messaging applications could be improved, in particular, concerning message deletion. Application developers could provide a notification where a message has been deleted from, or implement a dialog for explicit selection, to improve users' understanding of the capabilities of deletion functionality.

3.6. Conclusion

In this work, we studied users' preferences for the deletion functionality in instant messengers. We also investigated whether users could accurately determine from which conversation histories their messages were removed upon performing a deletion. We tested three different messengers (WhatsApp, Skype, Facebook Messenger) in a user study with 125 participants.

If deletion features were available, we saw participants use them in different ways, including editing messages. The majority of our participants preferred to be able to remove messages also from a recipient's device.

Deletion functionality in WhatsApp is different from the other two messengers in that users can explicitly select whether they want to delete a message on their local conversation history or also from the recipients' logs. We found that this led to a 30 % higher rate of correctly predicting the effects of deleting messages. We suggest that developers of other instant messaging applications describe the effect of message deletion more explicitly, e. g., by providing a dialog for selection as in WhatsApp, or include a notification indicating where the message has been deleted from.

4

Contractual Agreements for Data Revocation

Contents

4.1. Introduction

As users share more and more personal information with others online, the concepts of *digital forgetting* [119] and *revocation of online data* [184] become increasingly important for protecting their privacy. Once personal data is published, users are unable to continuously control their own information exposure in the respective online environment. Besides their difficulties with keeping track of all information they have made accessible to others in the first place, specifically the retroactive deletion of data after its initial publication is notoriously unreliable due to the distributed and open nature of the Internet.

4.1.1. Problem Statement

Many technical approaches to implement data revocation apply cryptographic erasure, i. e., publishing data only in encrypted form and making it irretrievable after expiration by a suitable key management [35,65,152, 154,166]. However, as we have learned in Chapter 2, such mechanisms do not adequately address users' needs to control their online information exposure. Instead, there is a need for more flexible solutions in terms of exposure reduction.

When we expand our view to a jurisdictional and also regulatory perspective, the European *Right to be Forgotten* [242] already received considerable attention in 2014 due to a ruling by the European Court of Justice (ECJ) [46]. The ECJ determined that online search engines need to provide an interface and procedures for EU citizens to request the removal of their personal information from search results. When we interpret this in a more general sense, there is an implicit obligation for online services to help users with controlling their online exposure in the long term. Thus, there is a need to further explore the scope of potential solutions for online data revocation that consider such dependencies between users and online services.

4.1.2. Contribution

In our work, we explore a mechanism for the revocation of online data that does not purely limit the availability of data in technical terms but provides monetary incentives such that providers take appropriate measures to support data revocation and to comply with expiration conditions demanded by users. In particular, we propose to conduct an agreement between a user, who owns a specific data item, and a platform provider, who offers access to the data. The agreement defines the expiration conditions for the data items. In contrast to existing approaches, our technique can be applied not only for data at the point of publishing but also for data that has already been made available in the past. Technically, we explore how smart contracts, as available in certain cryptocurrencies such as Ethereum [29], can be used to realize agreements between users and providers. Compared to other forms of reaching formal agreements, e. g., digitally signed PDFs, smart contracts allow a high degree of automation. Using a distributed ledger system is attractive, as it comprises a trusted third party in a decentralized manner. Our approach is designed to handle the majority of agreements on data expiration. Violations and disagreements in particular cases can still be handled in the jurisdictional system. Data owners might even refer to the contract as a piece of evidence in a potential legal action. In summary, our work makes the following contributions:

- We design a data revocation scheme based on contractual agreements that can be applied both to new data and to data that has been published in the past, effectively offering more flexible options to users. Users can take action both during the lifetime of data and after its intended expiration.
- We provide an extensive overview of the design space of solutions for the revocation of online data based on cryptocurrencies and contractual agreements.

- We provide insights into an instantiation of our protocol: We have implemented a prototype as an Ethereum smart contract to demonstrate the general feasibility of our approach.

4.2. Solution Overview

We will now provide an overview of the basic ideas of our approach, before giving detailed insights into our protocol in Section 4.3.

There are three main entities in the system: the *user* U, who initially holds and owns the data and wants to publish it online. The data should be published with a *provider* P, who makes it publicly available on a large *online platform* in the WWW, such as a social network, or image sharing website.

Whereas previous work has tried to enforce automatic deletion of data, we pursue an idea where both the user and the provider establish a contractual agreement. This agreement states that P will take measures to limit the distribution of the data (e. g., by enforcing a limited lifetime on the data) and that U is entitled to compensation if the provider violates this agreement.

For this process, we distinguish the following four actions: (1) registering the agreement, (2) publishing the data, and (3) probing if the data is still available. When a violation is recognized we (4) enter a settlement process. We illustrate the chronology of these actions in Figure 4.1.

Agreement between User and Provider First, the user and the provider need to reach a formal agreement. This agreement will include

(i) the data d it concerns, by means of a unique identifier,
(ii) the expiration condition on which the data should become unavailable, e. g., after a specific time t or an event e, and
(iii) the penalty p for the provider in case the agreement is violated by the data being available after time t or event e.

Similar agreements can be reached in the form of a traditional contract, but enforcing such agreements incorporating very small penalties, e. g., in the range of a few cents, is prohibitively expensive.

Publishing Data The actual publishing of data typically involves sending the data to the provider, who will make it publicly accessible. One interesting property of this approach is that Steps (1) and (2) can be initiated in arbitrary order. In other words, it is also possible to reach such an agreement even after the data was published. While most other approaches require deciding on using protective measures and even fixing specific parameters such as the expiration time upon publishing, our system can also be invoked retroactively.

Access Probing After publishing and before expiration by meeting the expiration time t or the event e, the data is freely available. Accessing the data requires no additional tools or measures. The user can easily probe whether the data is still accessible before and after expiration.

Settlement Process If the user detects a violation, i. e., the provider fails to handle the data correctly in making it available other than specified in the initial agreement, a penalty mechanism is triggered. A typical settlement could be a small financial compensation.

Adversary Model

The security notion for data revocation considered in previous works is security against a *retrospective attacker*: Basically, the retrospective adversary becomes active only after the data has been revoked. The attacker should not be able to reconstruct the original data after the expiration. This strong notion can typically only be achieved when the users are entirely honest and refrain from re-uploading the accessed data without protection to a different location. One can argue that re-uploading

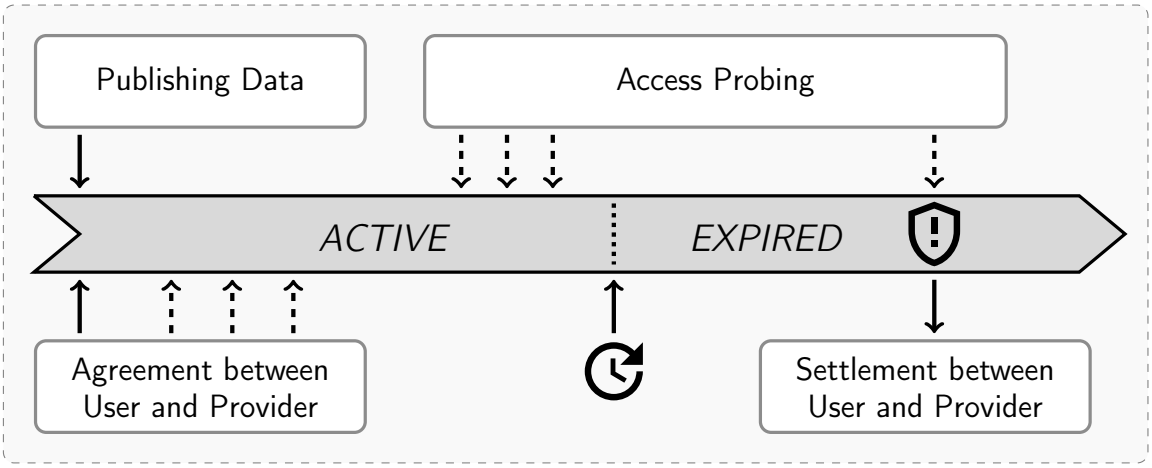

Figure 4.1. Timeline of our approach. User and provider conclude the agreement at the time of publishing or later. During its lifetime, the data is freely accessible on the provider's platform. A successful access after expiration leads to a settlement process.

the data while still available constitutes an explicit act of archiving, in which case the data should indeed be available over time [119].

Additional aspects consider how the protocol can be exploited by the participants to obtain an advantage over the opposite party. The use of financial compensations might tempt the user to claim violations in cases in which the provider complied with the agreement. The provider is therefore interested in a guarantee that the data have actually been accessible at the time of the claim. At the same time upon violation claim, the provider might be interested in false statements, i. e., contending that the data do not exist. Thus, the user is interested in a decision finding that can objectively determine whether the data exists and that is resilient to a provider tampering with it.

4.3. Revocation Contract Scheme

We use Ethereum smart contracts [29] as a convenient method to specify and validate agreements between the data owner and the service provider on the expiration of data. However, Ethereum smart contracts comprise certain limitations, such as accessing data outside the blockchain and

strictly enforcing payments between parties. For the specification of our protocol, we therefore require (i) the presence of a data feed for accessing external data and (ii) that a fraction of a potential penalty is placed as a deposit in the contract. The interactions in the stages of our protocol are illustrated in Figure 4.2.

1+2) Registration The registration process incorporates the first two steps as proposed in Section 4.2, since publishing data is a key subject of the agreement registration. A user U can publish a data item d along with an associated expiration condition exp_d on the platform of the provider P and will receive an identifier id_d for the data from the provider.

The user, who is identified by the address $addr_U$, can then initiate a transaction invoking the registration function of the smart contract C, passing id_d and exp_d to the contract.

The registration process is finalized as soon as the provider also invokes the contract, providing id_d, exp_d and $addr_U$. Along with the confirmation transaction, the provider places a deposit in the contract, from which a potential penalty can be paid out. We assume that the deposit can be significantly smaller than the prospective penalty, obviously separated for each provider since we expect that contract violations and the resulting settlement processes will only occur in exceptional cases. Deposits are not bound to specific data items, but instead, the contract balance should cover penalties for a certain proportion of all data registered in the contract. In practice there is no fixed order for the last two steps, i. e., it is also possible that the provider is the first party to pass the registration information to the smart contract.

During the registration process, the contract may check the conditions for general plausibility. In case the expiration condition is, e. g., a time t, it should be checked whether t lies in the future. The registration process does not necessarily have to be completed immediately after the publication of the data. It can generally be initiated at any time after the data has been published on the provider's platform.

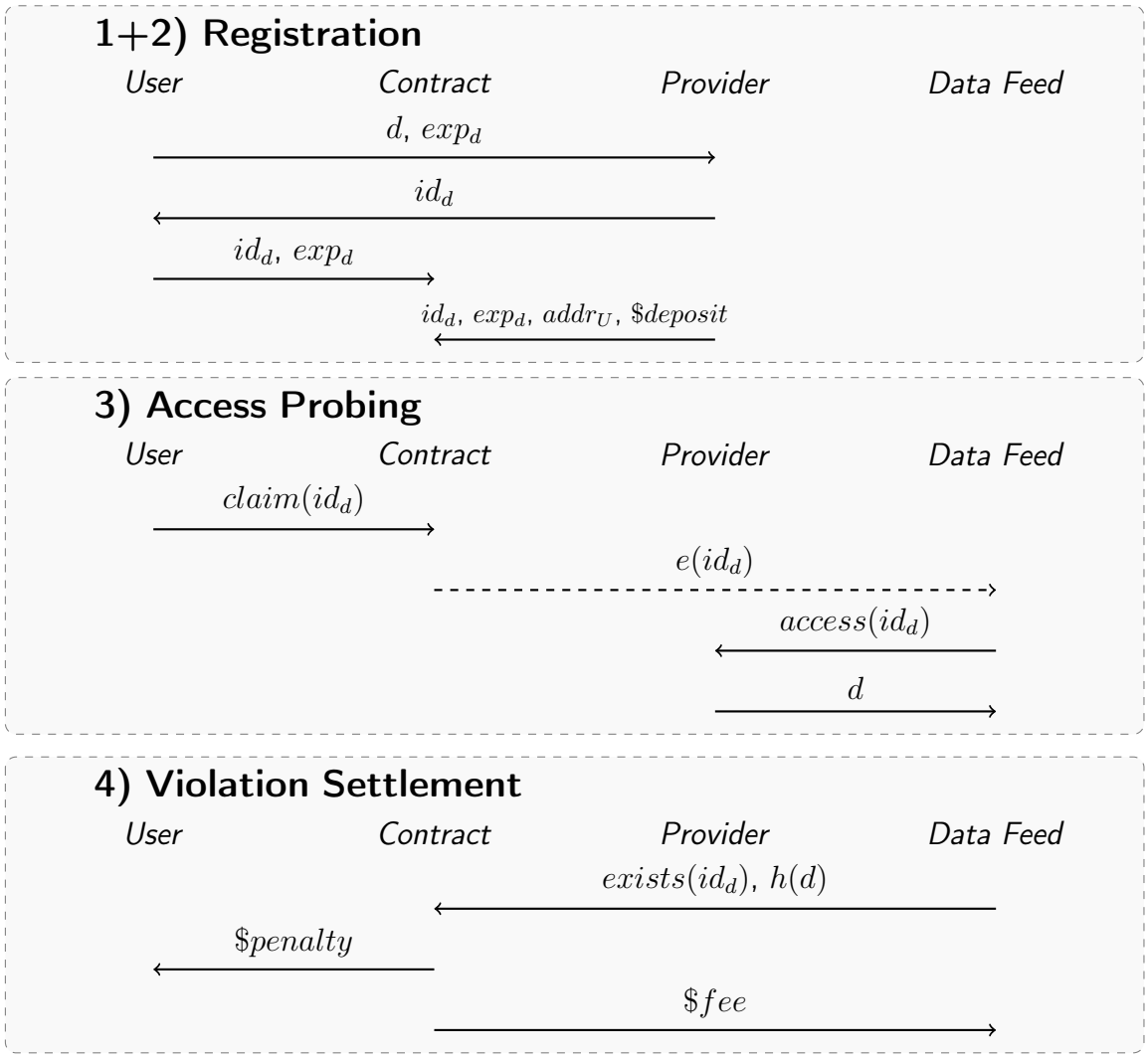

Figure 4.2. Protocols of the registration, access probing, and violation settlement processes.

3) Access Probing After its publication, the data is freely accessible on the provider's platform. There is no need to encrypt the data prior to legitimately reading it during its lifetime. In addition, the owner can probe access in direct interaction with the provider without the contract being involved. We can also delegate this task to a third party, e. g., the owner can assign the task of access probing to a trusted service. This requires that the data is publicly available during its lifetime – our scheme does not cover non-public data.

4) Violation Settlement Ideally, the provider would have revoked the data according to the expiration condition specified in the contract. Nonetheless, if the owner detects that the data is yet available, a transaction can be initiated to notify the smart contract about the violation, passing the corresponding identifier along with the function call. Violations can be reported not only by the user but by everyone who has access to the data. Thus, the user can even conclude out-of-band agreements at will with dedicated services to check the availability. These services regularly observe the provider's compliance with the contract and initiate settlement on behalf of the user in the case of a contract violation. When the contract receives such a violation claim, it must be checked whether the data is available online. Therefore, the smart contract will request an external data feed service S to retrieve the data from the platform of the provider. Finally, S will initiate a transaction notifying the smart contract that the access attempt has been successful and will provide a hash $h(d)$ of the retrieved data. As the last step of the violation settlement, the smart contract will pay out the penalty to the user and a small fee to reward the data feed service for taking part in the protocol.

4.4. Protocol Design Space

Next, we sketch how extensions of our protocol and alternatives in its design influence specific properties of the protocol.

4.4.1. Data Identification

In a basic approach, identifiers assigned by the providers are stored in the smart contract for data identification. Usually, this identifier does not change during the lifetime of data, regardless where it is referenced. To improve user privacy, the identifier should not be stored in plain text, but in a cryptographically secure hashed form. However, uploading the

data to an external website or even under a new identifier is sufficient to circumvent the protection.

Alternatively, data can be identified using hashes of the original data. Robust hashes [223, 247] are indented to tolerate minor modifications of data, such as re-scaling, compression artifacts, and similar in the case of images. The use of such alternative identifiers is generally feasible within our proposed protocol, it only requires the location information as additional input in the first step of the settlement process.

Such extensions appear more desirable for the user, but put a significantly stronger burden on the provider. Location-independent data identification raises additional challenges with regard to the accountability of contract violations. Moreover, by storing images while they are available and re-uploading them at a later point, a malicious user may be able to force compensation payments by the provider, even though the provider has behaved correctly. However, users are not entirely protected from malicious providers or third parties. If the script is transparently available, an adversary can gradually introduce slight changes into the data until they are classified as different to circumvent protection.

4.4.2. Data Feeds

Our system requires the presence of a data feed to incorporate real-world events from outside the blockchain. However, the actual request processing, contacting external resources such as a website, raises concerns regarding the trust required in these services and challenges in validating the delivered results.

First, the response delivered by the data feed should be correct, i. e., the data feed should process the data as they are present at the time of the request. Such a response must be time-bound, because it is not sufficient to show that the data under consideration have existed on the website at any point in time before. Second, transactions performed on the blockchain are publicly visible and irrevocable. The data feed cannot

simply access a website and write its contents (i. e., the data) to the blockchain, as this is effectively opposed to the goals of data revocation. We require the data feed to access the data from the provider website, perform an additional step such as determining whether the retrieved data comprise an image, and write the result to the blockchain.

Oraclize [158] and Town Crier [251] both use cryptographic means such as *TLS Notary Proofs* or *Software Guard Extensions (SGX)* to deliver a proof that certain data exist on a source website. From the perspective of the provider, such cryptographic attestations are attractive to prevent spurious claims from adversarial users. However, the use of a single service appears to be susceptible to malicious providers, as they could identify a particular data feed and deliver false responses to them to avoid the settlement process.

The concept of ChainLink [58] comprises multiple data feeds that access data independently from each other and are capable of performing computations on the retrieved data before responding. Its consensus mechanism for aggregating a result from the distributed responses (which can be built on trusted hardware such as SGX on an individual basis) reduces the trust required in the individual data feeds. Moreover, the use of a reputation mechanism that records false responses and a deposit-based penalty mechanism incentivize data feeds to deliver true responses. We imagine this as a crowd-sourcing approach in which even end-users can serve as verifiers in individual cases. Thus, massive misbehaving from the provider appears impossible, which ensures a high level of availability for such a system.

4.4.3. Complex Revocation Conditions

While the majority of prior work on data revocation simply use time-based expiration mechanisms, we suppose that smart contracts can be used to create almost arbitrary expiration conditions. However, these conditions need to be objectively observable through reliable data sources

providing appropriate interfaces. Such scenarios would require an additional step in the verification step of our protocol, whereas the general procedure would be similar and security requirements would not be stricter than for the data access verification.

In a naïve approach, the expiration condition is represented as an expression in natural language. There is a need for reliable information sources that are capable of processing the information in their represented form, e. g. providing an interface for natural-language processing. The trust in these services (e. g., Google, Wikipedia, Wolfram Alpha) can be reduced through the use of decentralized verification as proposed by ChainLink, assuming that the service cannot distinguish verification attempts from regular requests to their services. In addition, the stability of information must be taken into account, i. e., the verification must be resilient to short-time changes, e. g., malicious users manipulating the information to their advantage and then triggering the settlement process. This is feasible, if Wikipedia is used as a knowledge base, but can be prevented by also considering the information history.

However, with more complex expiration conditions, we also see the possibility that data items can become valid again after they have expired. Examples are scenarios in which data are supposed to be available only on specific days of the week, or have a daily access limit (under the assumption that the number of accesses can be verified reliably). This property makes the use of smart contracts a much more powerful instrument than previous approaches.

4.4.4. Financial Reserve Model

In order to guarantee the payout of penalties, the provider needs to place a deposit in the contract that is paid out when the contract is violated. However, in the case of large-scale application of our system, this would lead to large amounts of provider capital locked in the smart contract. Thus, we propose that only a fractional reserve (e. g., 1%) has to be

deposited in the contract, penalties are paid from the pool of all deposits, and that the provider has the possibility to withdraw money from the contract as long as the total amount is above the reserve threshold. This threshold is determined by the number of data items covered by the contract. If an item has expired and no violation has been reported for an adequate period of time, this item can be taken into account with less weight for calculating the threshold.

Large-scale violations that exceed the contract capacity determined by the total amount of deposits can still be handled resorting to the jurisdictional system. In this case, the contract can even be used as a piece of evidence to support that user and provider have agreed on the expiration of data beforehand.

4.5. Prototype Implementation

In this section, we describe our prototype implementation and evaluate the transaction cost incurring in its use and the scalability in terms of the numbers of data items that can be protected. Our contract employs time-based expiration conditions for data items that can be identified with a unique ID. We have implemented our prototype system using Ethereum with a local blockchain using the *Go Ethereum (Geth)* client. We have initialized Ethereum accounts representing a user, a provider, and the external service, as well as a smart contract in which items can be managed. For experimental interactions, i. e., registering or checking items with the smart contract, we utilized the *Ethereum Mist Wallet* application.

4.5.1. Smart Contract

We now describe the functionality of our smart contract implementation which we deploy on a local private blockchain for evaluation purposes. The registration process consists of two steps that can be executed in

arbitrary order. Both owner and provider have to commit information to create a valid entry. A user can register data in *addItem()*, providing as inputs its identifier and the remaining time it is intended to be available. The provider approves registration by using the function *confirmItem()*, also passing the identifier, the time left, and the owner's Ethereum address to the contract. We aim to ensure that the contract balance covers a minimum proportion of all registered data. After both parties have committed to the registration, the item agreement has become valid.

Data owners can be remove their data from the contract by calling the function *removeItem()*, which represents a cancellation of the agreement. Likewise, the provider can also withdraw confirmed items from the contract, as long as the owner has not added it.

The function *verifyAccess()* initiates the verification, whether data with a given identifier is available. In general, this function can be invoked by anyone and at any time, but the verification will only be triggered if the expiration date as stated in the record has been reached. However, access verification must be conducted by a data feed service external to the blockchain. For our prototype, we used an external script providing basic functionality. If the check is successful, the service can invoke the contract function *itemFound()*, which will initiate the compensation process. Thus, the contract will send the specified amount of Ether from its balance to the picture owner. In exceptional circumstances, i. e., if there is a widespread distribution of violations and many settlement processes are initiated at the same time, the Ether transfer may fail due to a low contract balance. The function *claimPending(id)* allows the user to initiate the compensation retroactively when the accessibility after expiration has already been verified before.

4.5.2. Cost Evaluation

The feasibility of our approach mainly depends on the transaction cost arising from the use of the contract and the number of agreements that can be achieved in total.

Transaction fees in Ethereum, referred to as *gas*, generally depend on the complexity of the transaction, but can also be specified by the transaction sender. If a lower fee is selected, it may take a longer time-span for the transaction to be processed. As of December 2018, waiting times for transaction processing have been 45 seconds [43] on average. This time-span seems acceptable, as our application is not time-sensitive in terms of a few minutes.

Table 4.1. Cost of contract execution.

Transaction	**Gas**	**USD**	**Actor**
Create	82,446	$0.0082	User
Confirm	35,107	$0.0035	Provider
CheckExpiry	27,199	$0.0027	User
VerifyAccess	28,621	$0.0029	User
ItemFound	22,071	$0.0022	Data Feed
ClaimPending	21,922	$0.0022	User

In Table 4.1, we illustrate the cost for the six transactions offered by our contract and the actor who needs to provide the fee on triggering the transaction. We assume an exchange rate of $100.00 per Ether and a transaction fee of 1 GWei. At the time of writing, miners of 30 % of total blocks have accepted this or even a lower fee [43]. However, these numbers are constantly changing, due to the current network load. Registering a new data item to the contract requires a user to provide a transaction fee of 0.8 cents. For a user who uploads one item per day, this results in a total amount of roughly $3.00 per year. For each item confirmation, the provider has to bear costs of 0.35 cents. The cost of the

other transactions will only incur in case of a contract violation. When we assume that the penalty for the violation is significantly higher (even in the range of a few dollars), the additional transaction cost will be negligible.

The amount of gas consumed by all transactions to be included in a new Ethereum block is limited to 8 million. When the contract registration requires 117,553 gas (see Table 4.1), and a new block is created every 15 seconds on average, this allows roughly 390,000 new registrations each day (2.75 million per week) under the assumption that the overall Ethereum network is not used for any other means. In 2017, Google has received roughly 2000 removal requests under European privacy law on a weekly basis [70], which is well below 0.1% of the limit that could be achieved within Ethereum.

4.6. Discussion

In this section, we discuss the effect of our proposal on trust assumptions, privacy aspects of practical implementations, and aspects of provider participation in our proposal.

4.6.1. Trust Requirements

Whereas we have introduced smart contracts for data revocation to reduce the trust required in providers when users upload personal data on their platforms, the use of data feeds still requires a certain level of trust. We consider the data feed a neutral adjudicator and, therefore, trust requirements in these entities are less critical than in providers who have – due to their business models – interest in making user content accessible as long as possible. However, trust in particular services is reduced by the presence of several alternatives users can ideally pick from, and also an approach based on crowd-sourcing. With regular other users serving as verifiers, it becomes harder for providers to make false at-

testations to data feeds to circumvent a violation detection, as regular accesses cannot be distinguished from accesses for verification purposes.

4.6.2. Metadata Privacy

The data accessible to smart contracts is stored on the blockchain, a distributed and publicly readable and irrevocable data structure. This may lead to two challenges:

First, the arising database might be a valuable target for an adversary who aims to collect all the sensitive data prior to expiration. This mainly depends on the number of data items covered by the contract and can particularly become a problem when our system is only used for sensitive data. However, if there is also sufficient non-sensitive data covered by the contract, which we advocate for, an attacker cannot directly reason that the data is sensitive just from its presence in the contract.

Second, the contract data can reveal information about the privacy preferences of individual users. If many data items registered in the contract belong to the same owner, it is evident that this user attaches importance to privacy in common and data revocation in particular. On the other hand, we can also conclude that this user also shares a lot of data publicly. We assume that the blockchain interactions involving a specific user provide pseudonymity, without direct linkability to a concrete person. Thus, information derived from the contract does not comprise an additional privacy leak.

4.6.3. Provider Participation

We see the use of smart contracts to register user data and specify expiration conditions similar to establishing a regular contract. Thus, a fundamental requirement is the willingness of service providers to support such a mechanism and active participation in the protocol. For a successful registration of a particular agreement, the provider must commit information to the contract. We cannot ultimately prevent that the

provider might refrain from entering the agreement in specific individual cases. However, such a misbehavior can be observed, as there will be open registration requests by users, which are publicly visible on the blockchain. This makes it possible for a user to check upfront whether a provider actually complies with the system, before finally uploading new data.

Recent studies [10, 127] have found out that users of online social networks are actively employing privacy preferences as offered by the providers for their data. Therefore, we assume that applying a service for data revocation as we propose can make social networks more attractive to users, especially to those who are generally more concerned about privacy issues. Moreover, our scheme can be utilized as a reputation mechanism, in that providers use the system to demonstrate that they take user privacy seriously as they comply with the preferences defined in the contract. If more providers apply the system, they can even compete with each other in reaching the lowest violation rate. From a regulatory point of view, our approach can be considered as a technical instantiation for establishing the users' right to be forgotten. In the future, technical revocation mechanisms, but also our proposal for contractual agreements can provide directions towards the automated handling of such removal requests. This seems desirable not only for the users but also for providers, in that it renders the manual check of revocation requests and dealing with individual cases unnecessary.

4.7. Conclusion

In our work, we have developed an approach for the support of data revocation. Different from previous work, we did not use cryptographic measures, but combined both technical and regulatory aspects in order to incentivize the provider to delete data as determined by its owner. Based on this idea, we have developed a protocol for the specification of revocation conditions in smart contracts and implemented a prototype

that supports time-based revocation conditions and is processed on a local Ethereum blockchain. The contract incorporates a penalty mechanism for data that remains available deviating from the conditions in the contract. With our approach, users can take action both proactively in defining expiration conditions for data they have published, and also retroactively in that they can get compensation in case the data provider has failed to delete data as specified.

Part II.

Usage-Driven Information Revelation

5
Preliminaries on Traffic Analysis

Contents

5.1. Motivation

In the first part of this thesis, we have seen a variety of challenges incurring when users deliberately share information with others in digital communication environments. We have learned about users' difficulties to retain information once they made it accessible to others, and that technical solutions to control or reduce their exposure do not adequately fit their needs.

For the remainder of this work, we shift our focus to information that users reveal to others when they use specific applications. The key difference compared to the cases we considered in the first part is that users do not share or reveal such information intentionally. Instead, we consider scenarios in which other parties conduct targeted malicious actions in order to retrieve information about the user. The fundamental principle behind this type of information revelation is known as *traffic analysis*, i. e., analyzing characteristics or patterns of the network traffic of a targeted user, induced by the applications they use.

While in earlier times of the Internet, the contents of unencrypted network traffic between clients and servers could be easily observed by external parties, the vast majority of web traffic nowadays relies on encrypted connections [155, 186, 219]. Standard web traffic is usually encrypted on the transport layer between client and server, mostly using TLS [103] or its designated successor QUIC [102]. On top of that, specific web applications can additionally encrypt their traffic on the application layer, with HTTPS for web browsing being among the most prominent use cases. In a messaging context, the Signal protocol is recognized as the gold standard for establishing end-to-end encrypted connections between messenger clients [42, 64, 88, 221], and has also been adopted by popular and widely used services such as WhatsApp. PGP, which was designed to add application-layer end-to-end encryption for emails, resembles a less successful example and has not managed to find wide adoption, mostly due to severe usability issues [239].

In encrypted connections between two entities, intermediate hops involved in the communication flow are unable to read the exchanged contents. Since virtually all modern digital communication environments use encrypted connections, traffic being encrypted will be a prerequisite for our subsequent analysis. In particular, having multiple layers of encryption is also a core property of the two applications we look into, i. e., the anonymity network Tor, and popular mobile instant messengers.

In this chapter, we first introduce the basic concepts of two types of well-studied traffic analysis attacks. Additionally, we provide an overview of work related to traffic analysis in general and in the light of the two end user information revelation scenarios covered in the second part of this thesis.

5.2. Traffic Analysis Attacks

The goal of traffic analysis is to gain information about Internet users and the services they use, despite web traffic being protected by multiple layers of encryption. The key technique for such attacks is observing network transmissions and evaluating features or patterns such as timings of packets in these transmissions. An adversary is usually interested in identifying the endpoints of a network connection, i. e., learning which client connects to which service.

Traffic analysis attacks are particularly interesting in the context of the anonymity network Tor [212] because it allows users to keep their privacy by concealing their online activities. By routing traffic through a set of *relays* – each of which cover traffic with an individual layer of encryption – it is not possible for an external observer to identify the target server a client communicates with. Likewise, on the last part of a connection through Tor on the server side, it is not possible to see from which client a connection originates. This inherent separation of identities, i. e., having no entity in the connection that can link its two endpoints, makes it necessary to analyze the network traffic in order

to gain insights into which clients and services communicate with each other.

De-anonymizing connections can be achieved in two different ways, each one resembling a class of traffic analysis attacks. In both types of attacks, the adversary monitors network connections and analyzes characteristics of traffic streams. Features of interest in network transmissions include timing properties such as inter-packet timings, or sizes and orders of network packets. In *Website Fingerprinting* attacks, the adversary focuses on monitoring client side connections in a single point and classifies the observed traffic streams using a previously obtained dataset of network traffic characteristics of websites. In *End-to-end Confirmation* attacks, the adversary monitors network traffic at multiple locations and aims to identify pairs of related transmissions, and thus, to learn which client is connected to which service.

Because Tor has become the most prominent target of traffic analysis attacks and is likewise its best studied use case from a research perspective, we introduce both attack types using Tor as an example. However, the underlying techniques used for traffic analysis can also be applied to other domains such as messengers.

5.2.1. Website Fingerprinting

Website Fingerprinting is a classification attack, in which an adversary aims to identify the website a client is accessing [79, 209]. To this end, the adversary first accesses a large set of websites, records the related network transmissions, and analyzes their characteristics to generate a fingerprint for each website of interest.

After this preparatory work, the attack can be conducted by monitoring network transmissions of a particular client. By comparing the respective traffic patterns with the previously recorded website fingerprints, the adversary can identify the correct website a client is accessing.

The success of such attacks largely depends on the extent and timeliness of the dataset. Since website fingerprinting is a classification task, a website can only be identified if its fingerprint is known as part of the pre-recorded dataset, and when the characteristics of the network transmissions remain consistent (which can be affected by, e. g., changes in structure or composition of the website) [92].

5.2.2. End-to-End Confirmation

In end-to-end confirmation attacks, an adversary aims to identify pairs of related transmissions that are not obviously linked with each other [48, 132]. This type of attack does not rely on previously collected data but requires access two sets of network transmissions, i. e., client- and server-side connections that are separated by an inaccessible network in between. The goal of the attack is to determine if a specific pair of streams, i. e., one from each set, and their characteristics and patterns are related to each other or not. In the case that both streams match, the adversary can determine that a specific user interacts with a particular online service.

In the context of Tor, the two sets of network transmissions typically reflect traffic streams collected at the entry and exit connections of Tor circuits. However, the concept can also be applied to other scenarios, such as a set of clients connected to the same or different servers of a messaging application, without directly seeing which pairs of clients are communicating with each other.

5.3. Related Work

We start presenting related work following the developments of the two types of traffic analysis attacks, i. e., Website Fingerprinting and End-to-end Confirmation. We then continue with research discussing traffic analysis attacks in Tor under realistic assumptions, and requirements for

such attacks. This is one of the key aspects of our contribution in Section 6 in which we analyze the feasibility of end-to-end confirmation in Tor in the light of their operational requirements. Finally, we consider research that explores traffic analysis attacks specifically in the domain of messengers, which represents the context of our contributions in Section 7.

Website Fingerprinting First approaches to systematic traffic analysis attacks date back to more than 20 years ago with first attacks on encrypted SSL connections and identifying the accessed websites by inspecting TCP packet headers [38] or analyzing counts and sizes of objects retrieved from the websites [209]. An attack by Hintz [79] focuses on connections through a proxy service used for additional encryption and anonymization purposes and is similarly successful in identifying the accessed website by evaluating the amount and sizes of transmitted packets. Subsequent attacks, also targeting proxied connections, conduct statistical analyses on more complex models with more features extracted from traffic patterns [26, 107].

Herrmann et al. [78] show that website fingerprinting attacks using statistical analyses such as Naïve Bayes classifiers directed at one-hop encryption proxies or VPNs reach classification accuracies of more than 95 %. However, these results do not directly translate into good performances on systems such as Tor – the same classifier only reaches 3 % accuracy due to perturbations in packet sizes and other characteristics when using Tor, rendering the attack much more challenging. The underlying data set is later reused by Panchenko et al. [149], who demonstrate that with using support vector machines for evaluation the accuracy of website fingerprinting in Tor can be massively increased. Cai et al. [32] propose an attack that can, in a closed-world setting, identify the correct website with up to 90 % accuracy from SSH-tunneled and Tor traffic, even with different application-layer defense mechanisms in place. In the context of Tor, Wang and Goldberg [227] propose the use of Tor cells

instead of TCP packets as a means to analyze traffic streams. Other attacks incorporate more sophisticated classification mechanisms such as k-nearest neighbor [226], random forests [77], or SVMs with updated feature sets [148], further improving accuracy and scale of website fingerprinting.

Most recent attacks rely on deep learning techniques to facilitate classification in website fingerprinting. While the first deep-learning approach applies neural network autoencoders to identify protocols from encrypted network traffic [233], Abe et al. [1] use the same technique for website fingerprinting in Tor. Rimmer et al. [170] present an extensive evaluation of various deep learning models applied to both new data and data used for several previous website fingerprinting attacks for performance comparison. Sirinam et al. [187] use a convolutional neural network architecture to conduct attacks on Tor traffic despite specific website fingerprinting defenses that Tor has put in place and successfully defeat these protection mechanisms, reaching more than 90 % accuracy against protected traffic. Rahman et al. [160] further improve this attack by proposing and developing a new set of packet timing features to be used for fingerprinting.

Besides these countless attacks, research has also proposed a number of defenses, particularly focusing on Tor and protecting its clients from the privacy threat induced by website fingerprinting attacks. Defenses against early website fingerprinting attacks are based on traffic morphing [245], i. e., making traffic look like originating from another website, modifying specific features such as the sizes and timings of network packets [116], or injecting dummy traffic [30, 54]. Cai et al. [31] systematically analyze website fingerprinting attacks directed at Tor traffic, evaluate which features of traffic are most important for attack success, and use their findings to develop a more efficient defense against website fingerprinting. Particularly in Tor, defenses arbitrarily increasing the volume of traffic such as padding or injecting dummy traffic constitute an unacceptable overhead due to latency requirements under Tor's

resource constraints. In this context, Juarez et al. [93] propose a probabilistic padding technique that produces less overhead and is similarly effective as previous defenses. Other proposals include application-layer defenses [39, 229] at both server- and client-side, i. e., instead of padding network packets they propose to modify website contents such as images or randomize the order of HTTP requests. In order to mitigate most recent deep-learning based attacks, Rahman et al. [159] modify traffic streams by means of adversarial examples, i. e., altering the traffic in a way that specifically leads to misclassification of websites by the learning algorithm.

End-to-End Confirmation The features used for end-to-end confirmation attacks as well as the proposed countermeasures largely overlap with those for website fingerprinting. Early correlation attacks against encrypted traffic rely on counting network packets [13, 181], or timing features of network traffic, such as timings between subsequently sent packets [231]. Proposed countermeasures against these attacks include mixing connections with timing delays [253] or dummy traffic [22]. However, correlation based on packet timings also works with mixed connections [48,104,185]. Danezis et al. [49] evaluate the feasibility of statistical disclosure attacks on mix networks, i. e., identifying communication partners from message frequencies, without considering specific traffic characteristics. Murdoch and Danezis [132] are among the first to analyze practical end-to-end confirmation in Tor by measuring mutual latency effects of concurrent traffic streams.

Many approaches to end-to-end confirmation evolve around watermarking, i. e., actively interfering with traffic in injecting specific patterns into one stream and re-identifying it at the other end. These techniques are mostly evaluated in the light of their applicability to general network traffic [81–83,108,168,169,230,249], while some specifically focus on their performance in the Tor domain [80, 109].

A persisting and continuously increasing challenge in correlation-based end-to-end confirmation attacks in Tor is the volume of traffic that needs to be analyzed, i. e., streams that need to be compared. In this context, Nasr et al. [138] develop an approach for compressing traffic features of interest to allow for more scalable attacks. Other works propose the use of deep learning techniques to increase both the volume of traffic that can be analyzed as well as the accuracy of attacks [137, 144].

Requirements for Realistic Attacks on Tor Whereas the technical capabilities of both website fingerprinting and end-to-end confirmation are continuously increasing, an inevitable requirement for traffic analysis attacks is access to the respective traffic streams. That is, an adversary must be able to actually monitor the traffic they are interested in at some point in the connection. As Murdoch and Zielinski [133] show, Internet exchanges managing traffic between networks of different Internet service providers, represent viable entities for traffic analysis with access to large fractions of Tor paths.

Specifically focusing on end-to-end confirmation, Johnson et al. [89] show that adversaries such as providers of autonomous systems or internet exchanges have access to both ends of individual connections of up to 95 % of Tor users within a three month period. In this context, several works discuss mechanisms for strategic path selection to protect clients and their traffic from being bound to individual strong actors such as autonomous systems [5, 16, 55, 94, 142]. Likewise, malicious interference with routing mechanisms as well as defenses against such attacks have been subject to analysis [210, 211, 213, 225].

For website fingerprinting attacks, Juarez et al. [92] provide an extensive evaluation of the practical feasibility of website attacks and their real-world requirements. They find that features such as attacker capabilities, or browsing behavior of users, are often simplified or even neglected, which leads to overestimating the accuracy of attacks. Their findings have been addressed in several subsequent works particularly

focusing on the practicality of such attacks, specifically regarding background traffic [228] or characteristics of evaluation data sets [188].

Shedding light on recently trending deep learning-based traffic analysis attacks, Rimmer et al. [171] identify several pitfalls in the design of measurement and evaluation setups facilitating correlation capabilities and re-evaluate previous attacks based on their newly developed systematic approach.

Our contribution in Chapter 6 focuses on access to traffic as an operational requirement for traffic analysis attacks. Complementing previous work measuring the capabilities of individual actors, we evaluate to what extent an adversary can actually determine in advance if they have access to traffic of a specific client. The ability to determine whether or not traffic analyses can be successful reduces the evaluation overhead for such attacks since unpromising targets can be omitted.

Traffic Analysis in Messengers Whereas the vast majority of traffic analysis research evolves around anonymity systems and Tor in particular, it can also be applied to other domains. This is particularly interesting for Chapter 7, in which we use traffic analysis techniques such as evaluating packet sizes and timings to reason about clients in messaging applications.

First attempts to analyze traffic streams of inter-personal communication target VoIP applications, and are able to identify the conversation language [244] or to detect specific phrases within the conversation [243]. Sengar et al. [180] use a watermarking scheme to trace streams of Skype traffic to identify to whom a specific client is connected.

Bahramali et al. [14] provide a broad overview of how information can be leaked through analyzing encrypted messenger traffic. Coull and Dyer [45] statistically analyze encrypted network traffic to identify the length of messages and their language sent with Apple's iMessage. Park and Kim [150] identify specific user actions within the Korean KakaoTalk messenger by analyzing encrypted traffic streams.

Our contribution in Chapter 7 complements existing research on traffic analysis in messengers in developing a concrete instantiation of such an attack that provides empirical evidence from end-to-end measurements under real-world conditions.

6

Operational Requirements for Tor Traffic Analysis

Contents

6.1. Introduction

With more than two million daily users, Tor [215] remains the most prominent anonymity system worldwide. Tor can serve everyday use cases with low-latency requirements and provides a fair amount of protection for user identities. However, this trade-off between performance and security comes at the expense of being vulnerable to traffic analysis attacks [67, 99]. Through the presence of such attacks in practice, users are facing the risk of unintended information exposure, which is a particularly severe issue in an environment they mainly use for privacy purposes. More precisely, users' identities can be de-anonymized and their online activities can be revealed by an adversary conducting traffic analysis attacks.

6.1.1. Problem Statement

Among numerous passive [48, 132, 137, 138] and active [81, 82, 168] traffic analysis attacks, we find convincing technical concepts approaching almost 100 % success rates for the de-anonymization of related streams [137]. At the same time, all of these attacks ignore the operational requirements for getting access to transmissions. That is, the attack can only succeed in case the adversary is able to monitor both endpoints involved in the connection. As Tor has a worldwide infrastructure of 6000 to 7000 voluntarily operated relays, this results in high resource requirements for monitoring candidate connections or nodes [142, 182].

In this context, *long-term* evaluations of end-to-end confirmation in practice have shown that adversaries controlling specific Autonomous Systems (ASes) or Internet exchange points (IXPs) can de-anonymize individual circuits of 100 % of users within a three-month period [89] and that compromise can be more effective with Border Gateway Protocol (BGP) level routing attacks [211]. However, the feasibility of end-to-end confirmation attacks on a per-case basis remains a blind spot, and

we must assume enormous resource requirements for a naïve monitoring and analysis of AS- or nation-state adversaries.

6.1.2. Contribution

In this work, we introduce three *stepping-stone* attacks that tackle the operational limitations of state-of-the-art E2E confirmation attacks and provide the adversary information about monitored connections as well as tools to interfere with the connection build-up procedure in Tor.

To remain in line with common attacker models in the context of traffic analysis attacks, we design our stepping stones in a way that does not introduce additional requirements or constraints for the adversary. To this end, we integrate our attacks into *defensive* features of Tor's circuit establishment procedure, making them a hard-to-counter "standard feature" of current Tor versions. This includes (i) inherent characteristics of the circuit establishment such as relay selection as well as (ii) mechanisms that have been introduced for protection purposes. For the latter, we focus on the `nTor` handshake ensuring *onion encryption* and *denial-of-service mitigation* that protects relays from being stressed. Figure 6.1 provides an overview of the systematic security analysis of these characteristics, which leads us to three stepping-stone attacks: *Exit Prediction*, *Circuit Replacement*, and *Multi-Target DoS*.

Exit Prediction provides additional information about the monitored connections, which helps to minimize the attack effort for non-global adversaries. For example, a nation-state adversary can conduct the Exit Prediction attack to check whether the exit traffic of a circuit passes through a country under control and, eventually, would lead to a successful E2E confirmation. This information about connections introduces a significant advantage over uninformed attacks in which all monitored traffic must be analyzed while related traces might not even be part of the monitored connections. In an empirical simulation study, we analyze the prediction capabilities of different probabilistic models and further

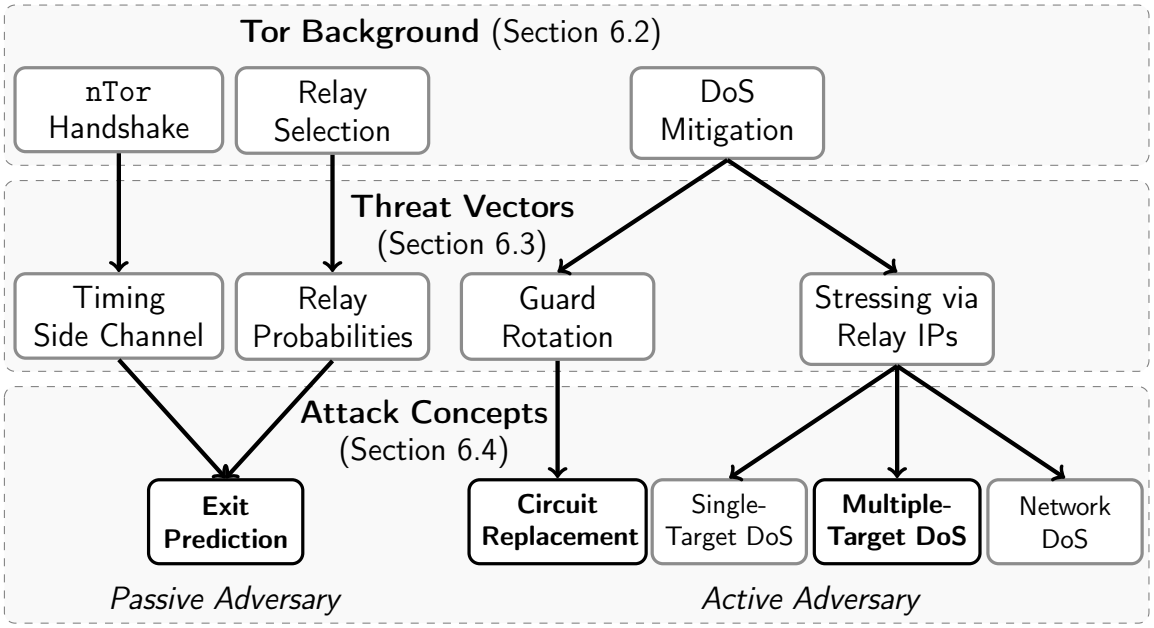

Figure 6.1. Structural overviews of threat vectors and attacks rooted in essential features and protection mechanisms related to Tor's circuit establishment procedure (top layer). We provide empirical case studies for three of the attacks (highlighted in bold, i. e., Exit Prediction, Circuit Replacement, and Multiple-Target DoS).

analyze how an adversary performs with and without the Exit Prediction stepping-stone. Our experiments show that, depending on the individual infrastructure of a country, the exit prediction ranks the correct relay of a circuit within the top 1 % to 18 % of all possible relays. This drastically reduces the required effort for an attack, as only a fraction of traffic needs to be analyzed.

Circuit Replacement and Multi-Target DoS both exploit the Denial-of-Service mitigation within Tor. Circuit Replacement allows an adversary to interfere with the guard set of a user by stressing the primary guard. This triggers the DoS mitigation and forces the user's client to switch to the next guard in the set, eventually introducing a new relay location and transmission path. This local application-layer routing attack allows an adversary to manipulate a circuit in case the original connection does not allow monitoring traffic. This introduces additional attempts to access the connection endpoints of a user. Our experiments show that the circuit

replacement provides an improvement of up to 33 % for adversaries that could not access traffic before the replacement attack.

Multi-Target DoS exploits the same DoS mitigation from inside the Tor network. Due to an implementation characteristic of the DoS mitigation, excessive connection attempts from *inside* the network are not blocked. This allows an adversary to stress single or multiple nodes in the infrastructure, which creates local failure or even complete intersections of network areas. Again, this can be used as a stepping stone for traffic analysis attacks, since it provides another tool to manipulate the connections within Tor. Our results show that individual relays can be disabled for one hour for around $ 20.

In short, the main contributions of our work are:

- We identify threat vectors rooted in core mechanisms and defensive features that are part of Tor's circuit establishment procedure.
- We analyze the characteristics and technical requirements for three attacks exploiting these threat vectors and facilitating traffic-analysis attacks.
- We use measurements of the live Tor network for simulation studies demonstrating the impact of the three mentioned attacks and their consequences for subsequent traffic analyses. Our experiments provide insights into case studies in real-world scenarios without harming real Tor users.

6.2. Tor Background

Connections through the Tor network use *circuits* that consist of three relays, i. e., an entry guard connecting to the user's Tor client, an exit relay connecting to the destination of the connection, and a middle relay as the link between the entry and the exit. The circuits are built during the bootstrap procedure in the client start-up of Tor and are ready-to-use for new connections.

In this section, we describe characteristics connected to the circuit establishment procedures in Tor, as well as defensive mechanisms that Tor has put in place to ensure user anonymity and to safeguard the stability of its network infrastructure.

6.2.1. `nTor` Handshakes

The circuit establishment procedure involves multiple layers of encryption. The Tor client conducts three key establishments handshakes with the entry, middle, and exit relays, as illustrated in Figure 6.2. To protect transmitted communication contents from relays in the circuit, the client follows the `nTor` protocol to establish individual layers of *onion encryption* with each relay separately. Since the client's identity must not be revealed to the middle and exit, connections with relays positioned later in the circuit transit through hop-wise encrypted TCP connections with previous relays. The `nTor` protocol ensures that exchanged messages remain secure while preserving the client's anonymity.

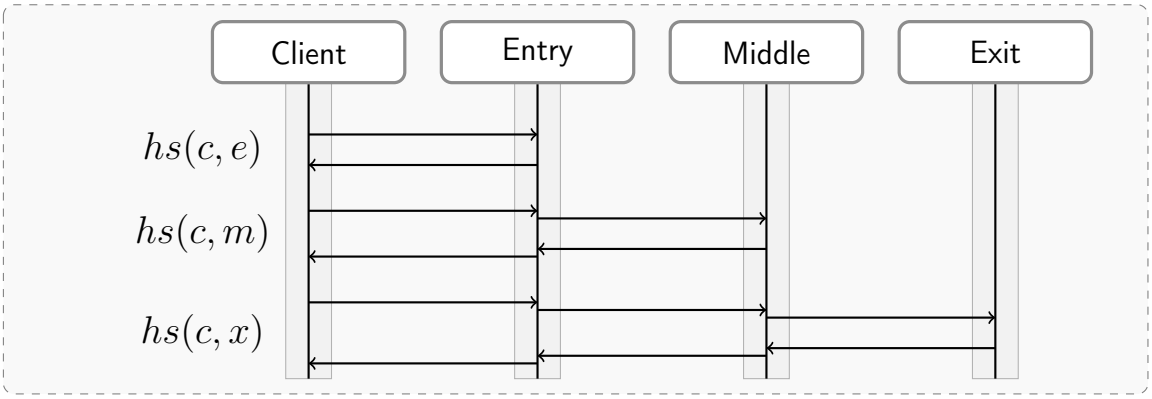

Figure 6.2. `nTor` Handshake. The client establishes keys with each relay in the circuit, in the depicted order. Key establishments with posterior relays transit through the circuit to keep the client identity private towards the middle and exit relays.

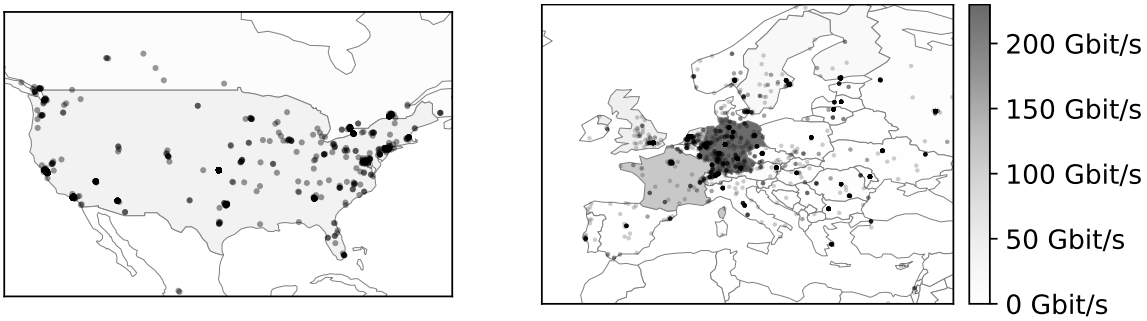

Figure 6.3. Distribution of Tor relays in North America and Europe. Darker areas denote higher total relay bandwidth (per country), relay locations are marked with dots.

6.2.2. Relay Selection

To keep track of all voluntarily contributed nodes, Tor uses a distributed consensus that consists of periodical votes on the existing infrastructure. This publicly available consensus assures easy access to the status of all available nodes. For a new circuit to be established, the client picks relays from Tor's worldwide infrastructure and focuses its choice mainly on the advertised bandwidth a node can offer in order to reach a fair distribution of traffic. The decentralized Tor infrastructure is backed by individuals and organizations worldwide, voluntarily operating relays with diverse amounts of resources they can commit. Thus, the geographical distributions of relays and the available bandwidth are subject to the capabilities of voluntary contributors. As illustrated in Figure 6.3, relay numbers per country are diverse with highest relay densities in central Europe (e. g., Germany, Netherlands, France) and the US. Especially in the Netherlands, there are 240 relays in an area of 500 square kilometers around the city of Amsterdam, providing 33 Gbit/s relay bandwidth in total, which comprises about 5 % of the overall Tor bandwidth.

There are further constraints that contribute to the composition of a Tor circuit by default, even though almost all of them can be manually overwritten by the client. By default, two relays from the same family (as specified in their descriptions) or residing in the same /16 subnet

(i. e., their IPv4 addresses must not be equal in the first two blocks) are not selected for a single circuit.

Guards and Exits The entry and exit relays in a circuit are particularly crucial for a circuit's security, as they directly communicate with the client (entry) and destination server (exit). The distinctive roles of both nodes are taken into account by assigning flags for relays that might serve in one of these critical positions of a circuit. While both flags are assigned after satisfying a series of requirements, entry guards are additionally organized in client-specific guard sets [76].

When a relay receives the `general` guard flag, the Tor client can sample it to become part of the client-specific guard set. Within these guard sets, the client keeps track of the connection status. This results in a sampled guard set of 15 relays on average, of which one to three relays have the status `up` and will be used in a circuit. Each of the `up` relays is assigned an index resembling the internal priority, i. e., the highest priority entry guard will be used in all general-purpose circuits if possible. The client switches to other primary guards of the set when the highest priority node is unavailable. The client creates a guard set once in the bootstrap procedure (if none is given) and updates nodes after a lifetime of several months [57].

The option to use guard sets can be changed by each client, allowing them to also use non-guard-flagged relays as entry nodes. In contrast, for circuits with traffic leaving the Tor network, only exit-flagged relays can be used in the exit position. The decision about allowing exit traffic is made by the relay provider. That is, relays are not picked at random but following a deterministic procedure.

Consequently, the actual composition of a circuit and its transmission characteristics depend on relay performance and geographical features. All information about available relays, their flags, or their advertised bandwidth is accessible from the consensus files that Tor updates in an hourly schedule using a decentralized voting infrastructure.

6.2.3. DoS Mitigation

One major threat to the Tor infrastructure are Denial-of-Service (DoS) attacks, in which the adversary floods relays through bursts of circuit and connection attempts. Since version `0.2.4.18-rc` [86] released in 2013, Tor implements DoS mitigation features that protect an entry relay from such excessive requests coming from a single IP address. We focus on DoS mitigation parameters targetting both the number of circuits that can be created from a single IP address and the number of parallel connections from a single IP address and defining consequences if the specified limits are exceeded (cf. Listing 6.1).

Listing 6.1: Denial-of-Service Mitigation Options

```
DoSCircuitCreationEnabled 0|1|auto
DoSCircuitCreationMinConnections NUM
DoSCircuitCreationRate NUM
DoSCircuitCreationBurst NUM
DoSCircuitCreationDefenseType NUM
DoSCircuitCreationDefenseTimePeriod N

DoSConnectionEnabled 0|1|auto
DoSConnectionMaxConcurrentCount NUM
DoSConnectionDefenseType NUM
```

The `Enabled` parameters define whether creating new circuits or establishing new connections is currently enabled. The `Circuit` options cover circuit creation requests, i. e., the creation rate, and the creation burst that define the allowed number of circuit creations per second and the maximum burst, respectively. The `MinConnections` defines the number of concurrent connections that must be present to trigger the mitigation feature and, eventually, the blocking of an IP address. For the `Connection` features, the maximum number of connections specifies the number of concurrent connections that are allowed from a single IP address at a time and closes new connections if exceeded. In combination,

the `Circuit` and `Connection` features block excessive requests and mark an IP address for the time defined in the `Defense` parameters. In case the relay provider does not specify any value for these features, the default setup still assures an active DoS mitigation.

6.3. Threat Vectors

In this section, we sketch how we exploit the presented characteristics and defensive mechanisms in Tor. The critical problem with the threats that we will present is that they are inherent to the fundamental concepts behind Tor, i. e., they cannot be mitigated without in-depth changes of the way how Tor works. For each threat vector, we describe how it is rooted within Tor and conduct preliminary experiments. This provides the baseline for the introduction of the three stepping-stone attacks.

6.3.1. Timing Side Channel

Similar to contexts like GPS, we can assume that the propagation time of signals between two nodes in a network relates to the traveled physical distance between these nodes. Given a suitable timing side channel, an adversary can make use of the timing relations and determine the geographical areas that relays are likely located in.

The cryptographic key establishment in Tor's circuit build-up procedure provides such a timing side channel. For such a circuit build-up, the Tor client and each relay in the circuit exchange messages as part of the `nTor` handshake protocol (cf. Section 6.2.1). Each message comprises a timing side channel. For example, observing the handshake between the client and the entry relay reveals the end-to-end round trip time between these two nodes.

In the following steps, we benefit from the fact that each new handshake message must follow the circuit infrastructure. More precisely, the handshake between client and middle includes the connection between

client and entry, of which we already know the individual *RTT(c,e)*. This enables us to approximate the transmission time between entry and middle relay:

$$RTT(e, m) = RTT(c, m) - RTT(c, e) \tag{6.1}$$

Following this principle, we can derive the transmission times of all three *individual* hops *RTT(c,e)*, *RTT(e,m)*, *RTT(m,x)* from the combination of timings.

We conduct a series of preliminary experiments to analyze the practicality of the handshake timing side channel to be later used as a stepping-stone for end-to-end confirmation attacks. In particular, we analyze how transmission characteristics depend on traveled distances, and we measure to what extent the cryptographic operations in the handshake protocol introduce overhead into the observable end-to-end timings.

Transmission Characteristics We analyze the propagation times of the empirical handshake data derived from 84,500 weighted circuit establishments (by weighted we refer to the standard Tor circuit build-up, i. e., we do not interfere with the selection of relays). Figure 6.4 visualizes the empirical handshake timing data ($n = 5000$, scatter) by distance between two hops approximated by a polynomial fit (lines). While we generally see higher handshake timings for longer distances, we also see timings scattered a lot around similar distances, with exit timings being higher (i. e., resembling lower transmission speeds) than entry or middle timings. The gap of 3000 km to 5000 km is caused by the static trans-Atlantic connection between the North America and Europe, e. g., inter-European or inter-US connections are either shorter (no transit) or longer (transit and distance to relays).

Static Overhead The observed end-to-end round trip times comprise transmission timings between the relays and also include a computational overhead for the key establishment procedures. We analyze this overhead

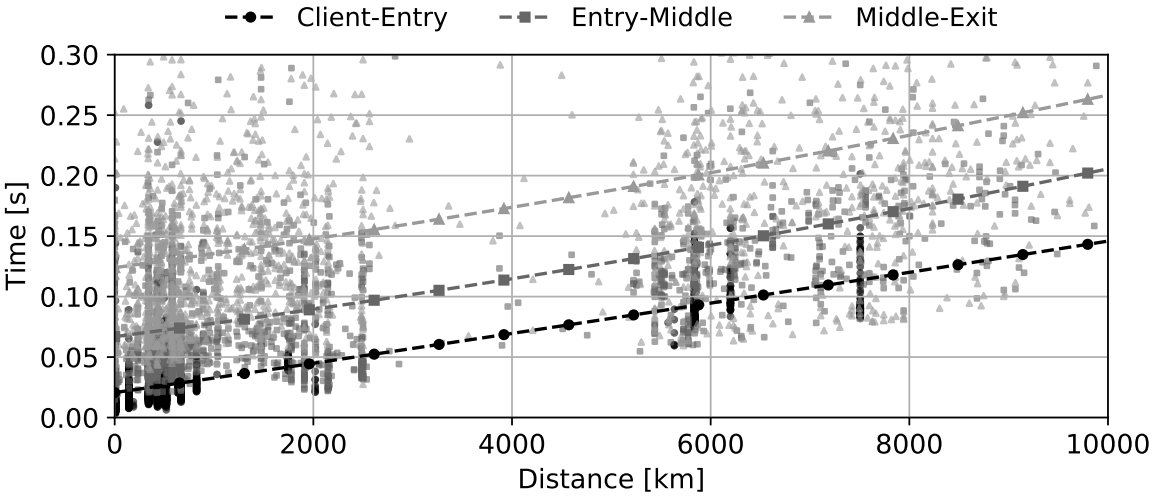

Figure 6.4. Distribution of handshake times by distance between pairs of hops. Data derived from a random sample ($n = 5000$) of circuit establishments measurements.

by running a patched Tor relay that records the time delta for *processing* the handshake, i. e., the time difference between the start and the end of the server handshake. Over a period of 12 hours we observed 138,000 key establishment timings on the server side with a median of 22 μs ($\sigma = 44\,\mu$s). As we see in the analysis of handshake timings, this overhead is negligible.

6.3.2. Relay Probabilities

When a Tor client establishes a new circuit, relay choices are not uniformly random but depend on several factors and are mainly driven by the advertised bandwidth of a node. Therefore, nodes have different probabilities to become a relay in the new circuit.

We describe two statistical approaches that can be used to assign probabilities on different levels, e. g., for individual nodes to become part of a new circuit, or for a circuit to contain relays located in particular countries.

Relay Selection-based Estimates The selection of relays in Tor is mainly driven by a node's advertised bandwidth, information that is

publicly accessible in Tor's consensus. Therefore, we can approximately determine the probability P_{bw} for a node x to be selected by considering its bandwidth as a fraction of the overall available bandwidth contributed by all nodes in the consensus:

$$P_{bw}(x) = bw(x) * \frac{1}{\sum_{i=1}^{n} bw(i)} \tag{6.2}$$

These calculations can be performed in real time and do not require any preparation other than downloading the hourly updated current Tor consensus.

Equally, we cannot only determine probabilities for individual relays, but also for groups of relays that are part of a specific autonomous system (AS) operated by a particular network provider, or relays located in a specific geographic area, each depending on the information available in the consensus.

When determining these estimates, we also consider constraints in relay selection, more precisely, we exclude all relays that are in the same relay family or in the same /16 subnet as one of the other relays in the circuit [52].

Timing-based Estimates We assume that, even if there is no direct relation between timings and traveled distances, transmission times between two specific relays remain similar over time, i. e., that we can observe the handshakes of a newly established circuit and identify the relays involved by comparing the handshake timings with previously collected data for the same or a similar connection.

Determining relay probabilities based on timings requires collecting a sufficiently large sample of timing data first, to determine the full range of likely timings for different connections. We can then generate empirical timing distributions for these connections, e. g., between pairs of relays or relay areas. Upon measuring the handshake time of a newly

established circuit, we can then extract a probability P_t for each relay x by considering the timing distribution for the respective connection.

While the overall idea here is conceptually the same as for relay selection-based estimates – assigning each node an individual probability for being involved in a newly established circuit – timing-based estimates require a lot more preparation.

6.3.3. Guard Rotation

A Tor client establishing circuits constantly uses the same guard relay in the entry position of all circuits, and usually does not change its behavior as long as the guard is available, i. e., changes it only in exceptional cases. This behavior is considered a security measure to protect the client and to reduce the risk of being exposed to malicious relays, since the entry connection is a critical point for client privacy.

The DoS mitigation features implemented in Tor relays (cf. Section 6.2.3) use client IP addresses as identifiers and do not allow any more connections or circuits from a specific IP, as soon as the limits specified for the mitigation features are exceeded. Since this mechanism is purely IP-based, it can also be triggered by excessive requests from entities pretending to possess a specific IP address. In this case, a client can be forced to switch from its primary guard to another relay from its guard set, without the change being necessary, and without the client being aware of the situation. Consequently, the client must create a new set of circuits.

We conduct a preliminary experiment to validate the presumed behavior of a client switching its main guard upon exceeding the DoS mitigation limits.

Triggering DoS Mitigation We verify the client's behavior by stressing our own Tor relay. We run our client on a local machine and set up our own relay `Torben` running on a remote virtual machine instance.

We first assure that `Torben` is the primary guard in the guard set of our client, such that it is picked as entry relay in the circuits we create. Upon starting the client, we drop all guards of the current guard set and manually add `Torben` to the empty set, rendering it the only guard. Shutting down the Tor client and restarting it adds new guards to the set. Besides our own relay, 2 to 3 additional entry guards with the status `up` are sampled in the list.

We run in total 20 Tor instances on our client, one of which uses the manipulated guard set with `Torben` as the primary guard; all others are used for stressing the DoS features. We first check our client's functionality, i. e., that it can build and use general-purpose and internal circuits. Both are satisfied when our relay shows up in the guard set, and Tor prepared a series of ready-to-use three-hop circuits with `Torben` in the entry position. In the next step, we build 20 circuits in each of the Tor instances with `Torben` in the entry position.

In the first Tor instance, all circuits of the initial build-up remain present and decay over time when their lifetime passes. It is impossible to build *new* circuits with `Torben` in the entry guard position. As older circuits disappear over time, Tor starts to build new circuits that now use one of the other relays of the guard set. These circuits show up in the circuit list and can also be used to attach streams for transferring data. Therefore, we have successfully triggered our first client instance to switch to another entry guard as a consequence of triggering the primary entry guard's DoS mitigation.

6.3.4. Stressing via Relay IPs

Even though Tor has established techniques to mitigate denial-of-service attacks, its mitigation features have one specific characteristic: They can only be triggered from *unknown* IP addresses, i. e., nodes that are not part of the consensus. For IP addresses that are part of the Tor network, DoS mitigation features, as described in Section 6.2.3, do not apply.

Therefore, targeted denial-of-service attacks affecting the stability and availability of Tor are successfully prevented when conducted from the outside but still remain possible from within Tor. However, exploiting this threat vector is limited to particular actors - for those who can either spoof valid IP addresses used by Tor nodes, or those who actually possess and control these address spaces.

Analyzing one exemplified consensus, we find 976 different Autonomous System (AS) operators that serve approximately 6700 relays of the Tor infrastructure. While many of the operators only serve 1 to 10 relays, larger AS regions include up to 746 (OVH SAS, 110 Gbit/s total bandwidth) or 409 (Hetzner Online GmbH, 100 Gbit/s total bandwidth) relays within their area of control. Consequently, without depending on additional hardware, a malicious provider can conduct a Distributed DoS attack using all IP addresses of relays falling into their AS area. In other words, the adversary can stress Tor relays without triggering their DoS mitigation, simply because IP addresses listed in the consensus are excluded from the mitigation.

6.4. Attack Concepts

Given the threat vectors of Section 6.3, we now introduce the specific attack concepts and how they support an adversary in conducting a end-to-end confirmation attack. To this end, we first introduce different models for the *operational* capabilities of an adversary. We then introduce the detailed concepts of the three stepping-stone attacks, which we later analyze in case studies of practical attack scenarios (cf. Section 6.5).

6.4.1. Attacker Models

In the context of network attacks, an adversary with access to transmissions on the Internet (IP) or Transport Layer (TCP, UDP) can conduct a series of active and passive attacks. The chance of being successful

mainly depends on the operational capabilities of the adversary. For example, a local adversary has access to the same type of information as a global adversary; however, the *amount* of information differs significantly. We specify three operational classes of adversaries that define the possible scope of an attack. For each attack concept, we extend this by specific technical capabilities.

Global Adversary. The global adversary can access all nodes in the network infrastructure and conduct arbitrary measurements.

Autonomous Systems and Nation States. An autonomous system can access all traffic routed through its service area. Depending on the centrality of a country's infrastructure, this can vary from multiple provider areas to one dominant provider operating the majority of connections. The nation-state adversary is an operational concept in which we assume a powerful entity that can request access to traffic in arbitrary points of a country.

Local Adversary. This adversary has access to traffic in a local network, e. g., uses the same access point in a public WiFi, and can monitor all traffic of this network.

6.4.2. Exit Prediction

The exit prediction provides the adversary with additional information about a connection. More precisely, we assign all relay candidates within the Tor infrastructure a probability for being in the exit position of a circuit. We do this by combining the timing side channel (cf. Section 6.3.1) with probabilities derived from consensus statistics. The outcome of an exit prediction is the list of all exit relays ranked in order of likelihood for being in the exit position of a single circuit. Upon receiving a relay ranking for a specific exit prediction, an adversary can determine whether subsequent traffic analyses are promising, depending on how the relays under their control (i. e., those they are able to observe) are ranked in the prediction. That is, the exit prediction can serve as an indicator of

whether an attempted end-to-end confirmation can be successful, eventually helping the adversary to save resources. It serves as an optional pre-analysis step that does not require additional technical or operational capabilities.

We first describe the concept behind and underlying assumptions of the attack and provide an empirical evaluation of the exit prediction using a simulation study in Section 6.5.1.

Technical Attacker Capabilities The technical and operational requirements for the exit prediction are included in all possible attacker models of an end-to-end confirmation: To exploit the timing side channel of a specific client, the adversary either requires access to the client's entry connection or to Tor relays that are flagged as Guard. This can be achieved by a locally restricted adversary, the minimum adversary for an end-to-end confirmation (cf. Section 6.4.1).

Concept Taking up on the *Relay Probabilities* as threat vector (cf. Section 6.3.2), the procedure for generating a bandwidth-based prediction ranking here is straightforward. The outcome of a prediction is simply a ranking of exit relays, with probabilities determined by their individual bandwidth fraction (cf. Equation 1). However, this is not a suitable mechanism to provide an adversary information on their chances for subsequent attacks. The prediction remains the same unless the consensus is updated and does not differentiate individual circuits and their characteristics.

The *Timing Side Channel* (cf. Section 6.3.1) refines the exit prediction based on the characteristics of a particular circuit. For simplification, we assume an adversary who can observe `nTor` handshakes from the entry position of a circuit to be established (e. g., by running a malicious entry relay). In this case, the adversary already knows the middle relay (since it is directly connected to it) and can observe the `nTor` handshakes between client and middle relay, and client and exit relay (without knowing about

the exit's identity). The transmission time between middle and exit relay can be approximated by the difference between the two handshake round-trip times (as observed from the entry position):

$$RTT(m, x) = RTT(e, x) - RTT(e, m). \tag{6.3}$$

From the transmission time between middle and exit relay, the adversary can determine probabilities for each exit relay candidate based on different propagation models. Those can be dependencies between transmission time and traveled distances or comparing the observed time with distributions of previously collected sets of transmissions times between individual relays, or between groups of relays. We introduce such specific propagation models and evaluate them as part of the exit prediction case study presented in Section 6.5.1.

Whereas we assumed that the adversary is located in the entry position here, it is also possible to transfer the concept to an adversary located between client and entry with access to the connection between them. In this case, the adversary must learn the middle relay's identity, which can principally be reached in a similar fashion. However, this two-step process adds more insecurity, since multiple relays may likely be in the middle position of the circuit, and likewise raise the efforts required, since the exit prediction must be conducted multiple times, i. e., for every likely middle relay. Exploiting the handshake timing side channel is thus independent of the attacker's exact position as long as there is access to some part of the entry connection.

6.4.3. Circuit Replacement

In this attack scenario, a client is forced to switch their primary guard and, therefore, to establish a new set of Tor circuits. Triggering the DoS mitigation in Tor in order to manipulate the circuit establishment process (cf. Section 6.3.3) leads to different routes taken between client and server, i. e. it comprises a routing attack on the application layer.

We consider a scenario where clients use standard three-hop circuits to anonymously access regular web services that are publicly accessible. More specifically, we do not consider the use of onion services, which is more complex in terms of circuit establishment.

The concept described here helps an adversary who aims to observe the end-to-end connection of a specific client, but who has learned that parts of the connection are unreachable. Forcing the client to update their circuit set can increase the adversary's chances since the new relays (and therefore, the connections) may be in areas under their control, i. e. increasing chances for successful traffic analysis attacks.

Technical Attacker Capabilities The guard rotation requires an adversary with the ability to spoof the IP address of a client, e. g., by using a TCP Man-in-the-Middle or by being located in the same NAT.

Concept The adversary acts as follows to enforce a client to switch their primary guard as illustrated in Figure 6.5. First, they monitor the client's circuit establishments to determine its primary guard which is used for all circuits by default. Subsequently, the adversary's goal is to trigger the DoS mitigation of the client's primary guard. Therefore, the adversary impersonates the client's IP address towards the guard (cf. Fig. 6.5 (1)). When the adversary has successfully triggered the mitigation, e. g., by establishing a sufficient number of parallel connections, or exceeding the limits specified for other features (cf. Listing 6.1), the client cannot use its primary guard any more (2) and is forced to establish new circuits using a different guard in its guard set (3).

This concept is not limited to a specific part of the connection an adversary is interested in. Specifically, it is not necessarily the entry connection that is under target. The idea behind this concept is that the client needs to generate an all new set of circuits, and therefore, resets the chances for an adversary who is unable to access a specific connection of interest. Thus, the guard rotation can be a means to gain access to a

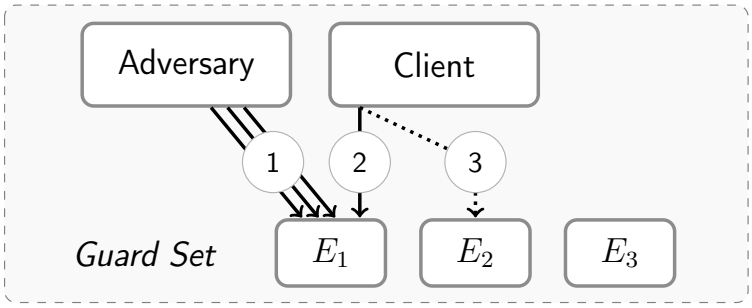

Figure 6.5. Exploit of DoS mitigation. The adversary stresses the DoS mitigation in the primary guard (1); the guard blocks the user's IP in response. The client cannot establish a connection anymore (2) and continues with the second guard in the set (3).

previously unobservable exit connection, as a means for end-to-end traffic analyses. The whole procedure described here remains stealthy, i. e., the guard cannot be accessed by the target client any more, but proceeds to operate normally for all other clients. From the client's perspective, the guard stops to operate and to accept new connections, but the client does not learn about the reason behind it, i. e., there is no reason for the client to assume being under threat.

6.4.4. Internal Denial-of-Service

Since all denial-of-service mitigation features in Tor only apply to unknown external IP addresses, the system still remains vulnerable to targeted denial-of-service attacks from inside. To achieve this, the adversary can transfer large amounts of traffic over Tor circuits such that the relays in this circuit reach their capacities. Such traffic must originate from Tor-internal IP addresses such as those used by regular relays. However, as Jansen et al. [87] show, denial-of-service can even be conducted from external addresses without triggering the mitigation, but depending on relay mitigation parameters, internal attacks from white-listed addresses always remain a fallback in the presence of tighter mitigation configurations.

Depending on the adversary's goals, denial-of-service affects Tor clients or the network as a whole at different dimensions, which we sketch with three examples at different scales. Directing a denial-of-service towards a single relay serves as an example to describe the general concept. We extend this by describing how scope and impact change when denial-of-service is conducted at larger scales, i. e., towards multiple relays at the same time or even targeting (large fractions of) the whole Tor infrastructure.

Technical Attacker Capabilities For an internal denial-of-service, an adversary must either be able to spoof IP addresses of Tor relays or actually possess and control these address spaces. This is eligible for ISPs or AS providers, i. e., this variant is limited to very specific and powerful actors. Alternatively, a weaker adversary can set up their own set of relays and start conducting the attack as soon as the relay(s) have become part of the Tor consensus.

Single-Target DoS Different strategies for denial-of-service against a single Tor relay have been shown to be feasible even from outside Tor without triggering DoS mitigation [87]. Strategies include establishing multiple circuits using the target relay and downloading large files to consume large amounts of bandwidth, or to include the target relay in a single 8-hop circuit multiple times. Therefore, we assume that such an attack scenario is realistic also from inside Tor with no DoS mitigation in effect.

Stressing a particular relay by generating large amounts of traffic renders the target relay unable to accept new connections for further circuits. This forces clients to include different relays in their circuits, thus, effectively re-routes client traffic. Target relays in such a scenario can be high-bandwidth guards or exits of interest that cannot directly be accessed by an adversary, e. g., one of the main guards or exits in a particular country. The adversary's goal is to increase chances for the substitute relay

to be located in an area under adversarial control, likewise for entry or exit traffic being routed through the area.

While motivation and eventual outcome are quite similar to the guard rotation strategy in Section 6.4.3 when it is directed against a specific client, the denial-of-service attack is different in that it affects all clients using the target relay at the same time. When the adversary cannot conduct the rather stealthy guard rotation targeting a particular client, stressing the relay may still be a fallback option. However, since the strategy requires a more severe intervention and effectively tears down the relay, it can also be observed easier.

Multiple-Target DoS Whereas the technical approach for the DoS attack against multiple targets is essentially the same as for blocking a single relay, the main difference is that there are multiple targets simultaneously. Likewise, the goal of re-routing Tor traffic through areas that can be easier accessed by the adversary remains similar.

Groups of targets can be, e. g., all relays in a specific (unreachable) AS, or all relays of a particular country. In Section 6.5.3, we take a closer look into the feasibility of such an attack scenario given the nature of the real Tor infrastructure.

Network DoS Further extending the denial-of-service to a large fraction of all relays drastically reduces the choice of relays that clients have to construct their circuits (and therefore, their overall anonymity set within Tor). Likewise, denial-of-service at large scale also affects the stability and reliability of the whole Tor infrastructure. Distributing the steady Tor traffic across a significantly smaller fraction of relays that, in turn, may not be able to handle the increased amounts of traffic can even amplify the attack, eventually cascading across the whole Tor infrastructure. The technical approach still remains the same – the adversary generates large amounts of traffic that exceed bandwidth capacities of

target relays. However, conducting the attack at larger scale simply requires more resources.

6.5. Case Studies

We present three empirical simulations as case studies for the Exit Prediction, Circuit Replacement, and Multiple-Target Denial-of-Service attacks. For each case study, we introduce the specific scenario, i. e., the empirical data the simulation relies on, explain how we evaluate the attack performance in this scenario, and present the results.

6.5.1. Exit Prediction

Combining `nTor` handshake timing data with relay distribution information allows assigning relays a probability for being used in a particular Tor circuit. In this section, we conduct a general empirical evaluation to analyze the feasibility of predicting the exit node of a Tor circuit in a practical scenario. In the next step, we evaluate to what extent the results of an exit prediction serve as a stepping-stone for end-to-end confirmation attacks. To this end, we analyze how the exit prediction reduces the otherwise immense overhead of processing the recorded traffic of multiple connection endpoints within Tor.

Evaluation Data Set To protect the security and privacy of real-world Tor users, we initially gather an empirical data set of Tor circuits that enables us to later *simulate* the exit prediction. Over one week, we record handshake timings with four different remote servers in New York, Amsterdam, London, and Frankfurt that act as Tor clients; we record two different types of circuits. First, *standard circuits* consist of relays that a client picks, i. e., they resemble the original selection criteria of Tor. Second, we extend the set of standard circuits by artificial *random circuits* that provide us with diverse transmission characteristics. Over-

all, we measure roughly 84,500 standard and 172,500 random circuits. We use this empirical data set for two main purposes:

(i) *Propagation Model.* As we predict possible exit locations from the monitored times of the circuit establishment handshakes, we depend on a realistic model of Tor's transmission characteristics. Such a model allows us to compare the measured times with general characteristics like propagation times and their relation to the traveled distance. Therefore, we use the empirical data set to derive a propagation model that we later use to estimate the target locations of exits. We use the generated data of roughly 257,000 handshakes to aggregate distributions of transmission times between pairs of countries the relays under consideration are located in. We generate probability density functions for transmission times between all pairs of countries. That is, we can determine a probability for a specific transmission time to have occurred in transmission between two countries. For the evaluation, we use 10-fold cross-validation, i. e., we compute 10 sets of empirical time distributions, each one leaving out 10 % of the *standard* circuits.

(ii) *Exit Prediction Simulation.* The monitored circuits serve as a test set for the exit prediction simulation. We randomly pick 10,000 standard circuits and predict the exit relay for each of them. Since we know the correct relays, we can use this information to measure the quality of a prediction. When we predict the exit of a standard circuit with the propagation model described above, we ensure to always use the model instance that the circuit under consideration was not included in (*cross validation*).

Evaluation Metrics and Assumptions For the exit prediction, we generate and compare four different probabilistic relay rankings described here. First, we consider a bandwidth-based (**BW**) prediction that simply assigns a probability $P_{bw}(x)$ to each exit x depending on its bandwidth fraction. Second, we consider a (**TIME**) prediction based on `nTor`

handshake timing information. Given that the location (country) of the middle relay is already known (cf. Section 6.4.2), we can determine a probability $P_{time}(x)$ for each exit relay x by considering its country and look up the likelihood for timing value in the distribution for the corresponding country pair (middle, exit) in the propagation model described above. We also consider a combined (**COMB**) prediction that takes into account both probabilities in combination. Since both observations are independent of each other, we determine the combined probability $P_{comb}(x)$ as follows:

$$P_{comb}(x) = P_{bw}(x) \cdot P_{time}(x) \tag{6.4}$$

Solely for reference, we also provide results for a prediction with all relays ordered randomly (**RAND**). However, ranking all relays in random order is not a realistic strategy. Since relays with higher bandwidth are more likely to be picked for a circuit, we consider a bandwidth-based ranking the baseline strategy for a strategic attacker.

When evaluating the accuracy of the four different predictions, we aggregate the relays by country. This aggregation results in sets of relays, each of which a potential nation-state adversary is able to observe. We focus on nation-state adversaries, as each country has its own concept to treat Tor traffic. This results in individual jurisdictions where all traffic through the country experiences the same "treatment", e. g., legal regulations that consider Tor traffic as suspicious will allow the monitoring of transmissions. Considering a nation-state adversary allows us to predict the consequences for potentially malicious key countries of Tor's infrastructure.

For each nation-state, we consider the median relative predicted rank across all cases in which the circuit's true exit was located in the respective country. As an example, for all circuits whose exit is located in Germany (DE), we denote the median relative rank of the true exits across all predictions for these circuits.

Table 6.1. Median relative ranks of the true exit across all predictions.

	DE	**US**	**FR**	**GB**	**CH**	**NL**	**AT**	**SE**	**RO**	**CA**
COMB	4 %	12 %	7 %	9 %	10 %	8 %	1 %	6 %	11 %	18 %
TIME	10 %	25 %	13 %	15 %	18 %	17 %	7 %	16 %	25 %	18 %
BW	11 %	21 %	16 %	23 %	22 %	22 %	1 %	13 %	18 %	35 %
RAND	49 %	50 %	50 %	51 %	53 %	50 %	49 %	50 %	48 %	52 %

For each country, we further evaluate how the outcome of an exit prediction can help an adversary in conducting end-to-end confirmation attacks more strategically. To this end, we evaluate how the exit prediction is a stepping-stone to successful and resource-efficient traffic confirmation as the adversary only attacks a specific fraction of the exit prediction ranking. To obtain these results, we simulated the exit prediction for 10,000 circuits randomly picked from our evaluation data set.

Results Overview Table 6.1 presents the exit prediction performance. The results show the median relative ranks of the true exit across all predictions sorted by exit country. The combined prediction (COMB) achieves the most accurate results across all countries; we provide the results of a randomized prediction (RAND) for reference. We focus on the top 10 countries w. r. t. to their total exit bandwidth. A lower value indicates a better ranking of the actual exit in the prediction, thus, a higher prediction accuracy. Due to the skewed distribution of resources within the Tor infrastructure, the results for the prediction models vary across different countries.

General Performance We now compare the results in the US and Germany (DE), two essential countries for the Tor infrastructure in the number of relays and the bandwidth they provide. In the median case, a relay located in Germany is ranked in the top 10 % with the time-based exit prediction and in the top 11 % using the bandwidth-based prediction. When combining the two approaches, any relay located in Germany is ranked in the top 4 % of all relays in the median case. The

US appears to be an exception, with the prediction performing worse than for other countries, particularly w. r. t. the time-based prediction. We attribute this to different geographical circumstances, e. g., significantly longer transmission distances than across the other countries located in Europe. However, combining the timing and bandwidth ranking still provides a median ranking within the top 12 % of relays. For the remaining countries, we see similar performances for both individual prediction metrics, with the timing-based prediction performing slightly better. Combining the two approaches improves the performances for relays located in all countries under consideration.

Stepping-Stone A nation-state adversary operating in a particular country can use the outcome of the exit prediction to act more strategically and reduce its efforts for subsequent traffic-analysis attacks targeting particular Tor circuits. We now analyze how successful one exemplary adversary (US) can be in end-to-end confirmations and how much exit traffic they need to monitor when they only monitor their relays ranked above a specific threshold in the exit prediction ranking. For reference, an adversary without assumptions about the actual exit (*baseline*) would always monitor *all* of their relays and analyze traffic in decreasing order of relay bandwidth until their end-to-end correlation has been successful (i. e., implicitly follow the bandwidth-based ranking), since relays with higher bandwidth generally have a higher likelihood to be picked for a circuit.

Figure 6.6 illustrates (a) what fractions of accessible traffic the adversary can observe (i. e., their expected success rates, y-axis) when they only monitor relays ranked within a specific fraction of the prediction (x-axis), i. e., above a specific *threshold rank*. While (a) directly connects the success rate to the outcome of a prediction, it does not consider the adversarial effort required to monitor all relays above the threshold in the prediction. Accordingly, (b) shows the fraction of relays the adversary monitors, i. e., what fraction of their relays is ranked above the thresh-

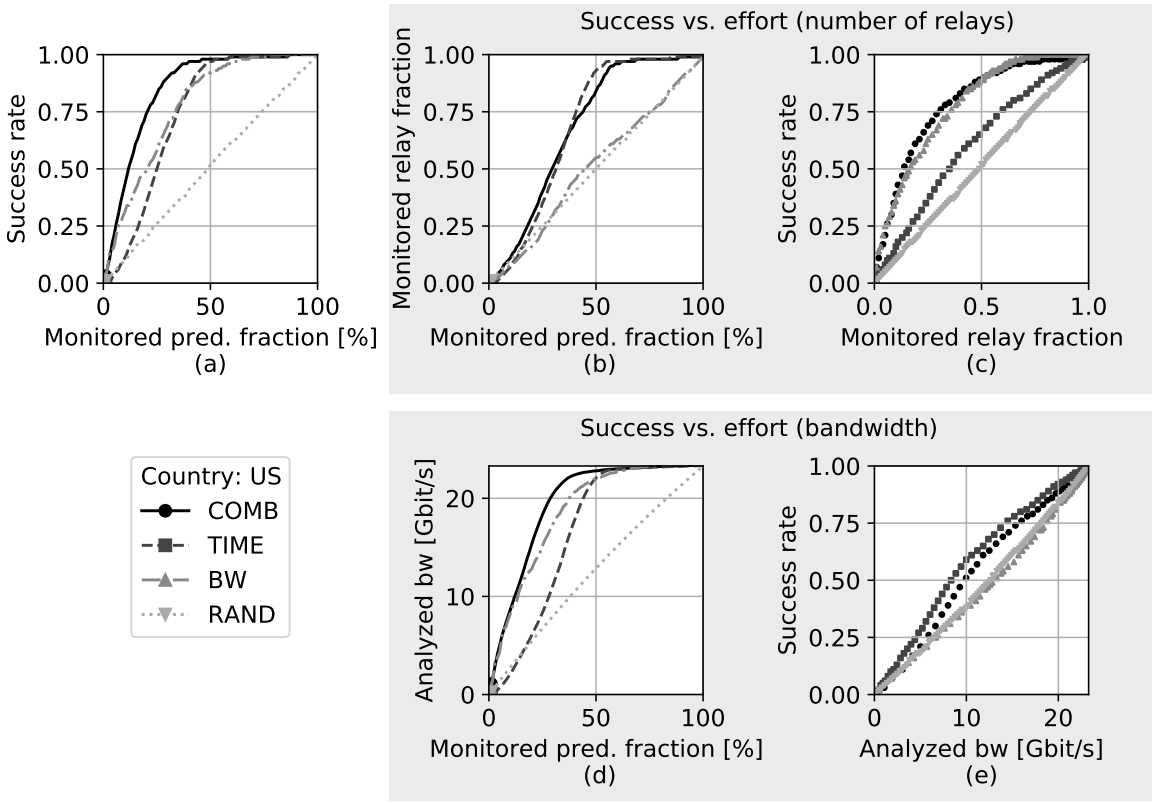

Figure 6.6. Detailed exit prediction performance evaluation for the US adversary. (a) shows the relative success rate, i. e., what fractions of accessible traffic an adversary can observe when monitoring relays within a specific fraction of the prediction. The next two blocks (each highlighted in grey) compare success and adversarial effort in two steps. While (b) shows the fraction of relays under adversarial control within a specific prediction fraction, (c) combines (a) and (b) by showing the success rate relative to the monitored relay fraction. Likewise, (d) shows the bandwidth to be analyzed when monitoring a specific fraction of the prediction, and (e) compares the analyzed bandwidth with success rates.

old, and (c) combines (a) and (b) showing how the success rate depends on the monitored fraction of relays. In the same way, (d) and (e) connect monitored prediction fractions to analyzed relay bandwidths and to success rates in two steps.

The bandwidth that needs to be analyzed (depending on the monitored ranking fraction) is determined as follows: The adversary analyzes all traffic streams following the order in which they are ranked in the prediction. In case the adversary is *not* able to access the exit traffic that relates to the entry point of the connection, i. e., both parts of the end-to-end confirmation, the traffic of all relays above the threshold rank needs to be analyzed (without success). In case the adversary *is* able to access exit traffic, only the traffic streams of the actual relay and all relays ranked *higher* in the prediction need to be analyzed. There is no need to continue analyzing the rest of the monitored traffic streams as soon as the end-to-end confirmation has been successful. We assume that the underlying analysis technique is able to reliably distinguish between related and unrelated streams, i. e., a correct match of related traces always leads to a clear result.

As we can see in Figure 6.6 (e), the timing-based and combined prediction achieve higher success rates per analyzed bandwidth than following the bandwidth-based relay ranking of the baseline adversary. An adversary who analyzes 10 Gbit/s of exit traffic achieves a success rate of 39 % when analyzing their relays in decreasing order of bandwidth. When monitoring the `nTor` handshake and ranking relays based on these timings (or in combination with bandwidth information, respectively), the adversary can achieve a success rate of 59 % (timing-based ranking) when analyzing the same amount of traffic.

Table 6.2 corresponds to the US results in Figure 6.6 (e) and lists AUC (Area Under the Curve) values for the success rates, where larger values indicate a better performance. We use the AUC to summarize the overall performance of a nation-state. As we see, the bandwidth-based prediction (BW) achieves a similar performance (in terms of success per

analyzed exit bandwidth) as the randomized exit prediction (*RAND*) in all countries. The time-based prediction (*TIME*) and the combination of timings and bandwidth (*COMB*) achieve higher performances across all countries, e. g., the timing-based prediction performance in the US is 27 % higher than the bandwidth-based prediction. Please note that these numbers can only be compared between predictions *within* the same country due to the different exit bandwidth amounts across different countries.

Table 6.2. Attack success rates per analyzed exit bandwidth (AUC).

	DE	**US**	**FR**	**GB**	**CH**	**NL**	**AT**	**SE**	**RO**	**CA**
COMB	2.64	12.61	1.53	2.29	3.84	3.66	4.72	1.26	1.54	1.92
TIME	2.86	13.74	1.48	2.51	3.94	3.95	6.11	1.29	1.58	1.91
BW	2.31	10.81	1.41	2.14	3.48	3.30	4.23	1.06	1.31	1.45
RAND	2.33	11.02	1.48	2.34	3.52	3.30	4.91	1.11	1.35	1.44

Our results imply that strategically analyzing exit traffic based on observed handshake timings can actually increase the adversary's success for end-to-end confirmation attacks, or reduce their required efforts, accordingly. That is, the adversary has a choice in the selection of the ranking strategy and, consequently, the trade-off between accuracy and analysis overhead.

6.5.2. Circuit Replacement

We now evaluate the impact of the circuit replacement attack by simulating how nation-state adversaries can improve their chances of observing Tor exit traffic by enforcing the guard rotation directed against a specific client.

Table 6.3. Improvement through circuit replacement.

DE	**US**	**FR**	**GB**	**CH**	**NL**	**AT**	**SE**	**RO**	**CA**
5 %	33 %	4 %	6 %	8 %	7 %	15 %	3 %	3 %	5 %

Evaluation Data Set We use our experimental setup as described in Section 6.5.1 to generate test sets of Tor circuits as they are pre-built in a client-side Tor instance. For each test set, we consider a single guard node in the entry position of all circuits and add up to 1000 circuits to the test set. We only take into account guard relays for which we have a minimum of 50 circuits with that relay in the entry position, resulting in 37 different guards to consider.

Evaluation Metrics and Assumptions We consider the fraction of circuits with the exit node located in an adversarial area within each test set, representing a client's updated guard use after a circuit replacement attack has been conducted. Given that an adversary cannot access the exit traffic in a particular circuit, these numbers represent the chance of accessing the traffic after the attack has been conducted.

We define adversarial areas on a per-country basis, focusing on the top 10 countries w. r. t. to their total exit bandwidth.

We assume that clients behave regularly and only use circuits with their primary guard in the entry position and only switch to circuits with their secondary guard when the primary guard becomes unavailable. Furthermore, a client has a usage profile that puts similar loads on all available circuits, i. e., we consider all circuits with the same entry guard equally.

We now provide our results of the application layer routing attack simulation on a per-country basis.

Results The circuit replacement can be used for additional attempts to gain access to both ends of the circuit of a specific user. That said, an adversary conducts the attack in cases where *no* access to the exit relay or traffic is given. Table 6.3 summarizes the achievable *improvements* for nation-state adversaries who can access the traffic of all relays within their area. The numbers represent the average fractions of circuits with exits in the respective country after the circuit replacement.

One example of a substantial improvement is the US. On average, one third of exits are located in their area after conducting the circuit replacement. The improvements we report in Table 6.3 roughly match the bandwidth fractions of all exit relays located in the respective countries as they appeared in the consensus used for the simulation. This is a plausible outcome since relays to be included in a circuit are mainly selected based on their bandwidth. However, these numbers are subject to constant change, depending on the evolution of the Tor infrastructure and changes in bandwidth distributions. As of March 2021, the bandwidth fraction of exit relays in the US has dropped to 20 %; the fraction of exits in DE has increased to 31 % with (presumably) equal consequences for the attack success.

6.5.3. Multiple-Target DoS

We evaluate the impact of internal denial-of-service directed at multiple target relays. We consider the bandwidth cost required to stress the relays under target and how the attack can improve the adversary's chances in a traffic analysis attack scenario.

Evaluation Data Set Our evaluation is based on a single Tor consensus (as of 23 October 2020) supplemented with location data and AS information retrieved from an IP geolocation service. The consensus contains 6735 relays in total, 3717 of which have a guard flag and 1427 can be used as exit nodes.

Evaluation Metrics and Assumptions In order to evaluate the practicality of a multi-target denial-of-service attack in Tor, we assume that a relay can be effectively stressed by generating the amount of traffic that corresponds to its assumed link capacity. We refer to this as *DoS bandwidth*. Following the approach of Jansen et al. [87], we consider the relay bandwidth as advertised in the consensus and estimate its link capacity

as the next higher value in a set of fixed bandwidth classes (1M, 10M, 100M, 200M, 500M, 1G, 10G [bit/s]).

We consider DoS bandwidths on a per-country basis, resembling a scenario in which a nation-state adversary with access to all relays in a particular country aims to increase their chances for accessing Tor traffic by targetedly disabling relays in areas out of reach.

Finally, we estimate the cost to perform such an attack by taking into account the amount of traffic that is required to stress a relay with a given link bandwidth over a specific period of time.

Table 6.4. Relay bandwidth vs. required DoS bandwidth.

Country	**Relays**	**Total BW [Gbit/s]**	**DoS BW [Gbit/s]**
Guards			
Germany	861	208.92 (35.6 %)	1228
France	561	105.15 (17.9 %)	420
United States	615	46.65 (7.9 %)	87
United Kingdom	175	45.25 (7.7 %)	197
Netherlands	227	44.70 (7.6 %)	223
Exits			
Germany	289	78.98 (35.7 %)	305
United States	355	30.38 (13.7 %)	50
France	126	27.42 (12.4 %)	60
United Kingdom	103	26.27 (11.9 %)	53
Netherlands	63	11.97 (5.4 %)	52

Results The DoS bandwidth required to stress all relays varies across different countries, depending on how much bandwidth relays in these countries provide and how the bandwidths are distributed across all relays in a country. In Table 6.4, we present required DoS bandwidths for stressing all guards and exits in the top 5 countries w. r. t. to the provided bandwidth.

We see that guard bandwidths are higher than exit bandwidths across all countries, therefore also requiring higher DoS bandwidths when tar-

geting guards. We also observe that higher total bandwidths per country do not directly translate into higher DoS bandwidths. When considering the guard bandwidth per country, the US, GB, and NL provide roughly 8 % of the overall guard bandwidth each but require different amounts of DoS bandwidths to be stressed. These differences can also be seen when considering the required DoS bandwidths per individual country for Guards and Exits (cf. Figure 6.7).

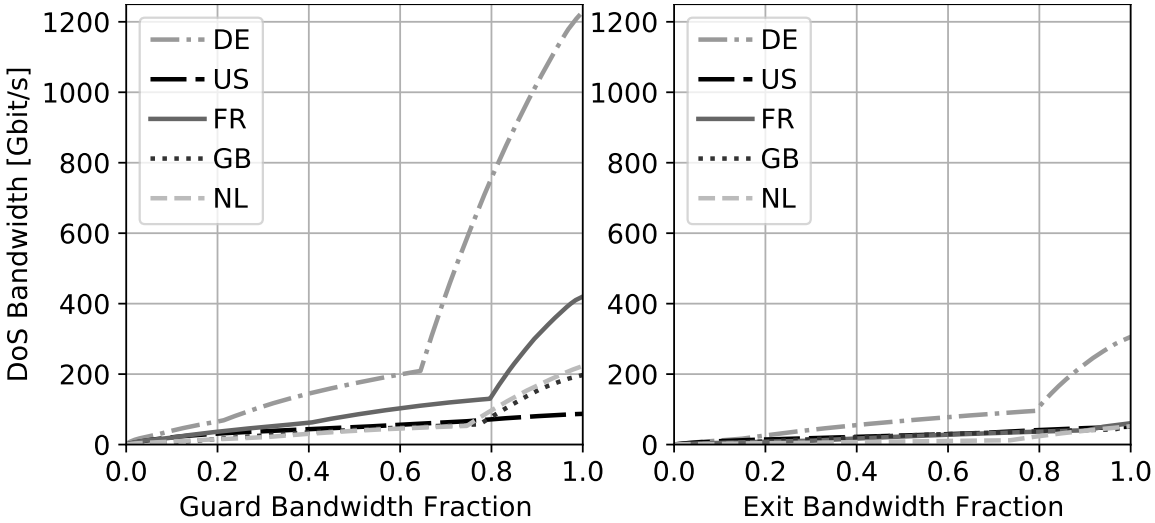

Figure 6.7. DoS bandwidth cost for guards and exits per country.

For example, stressing 80 % of the exit bandwidth in DE requires roughly 100 Gbit/s of DoS bandwidth, whereas stressing the remaining 20 % additionally requires 200 Gbit/s of DoS bandwidth, i. e., twice as much added on top. The reason for this is the different distributions of bandwidths within these countries. For all European countries (DE, FR, GB, NL), we see that there is a small fraction of relays individually contributing high bandwidths of up to 1000 Mbit/s, and in some cases even above. Due to the assumption of considerably higher link bandwidths in this case, a small number of high-bandwidth relays largely influences the overall required DoS bandwidth in these countries. In contrast, such high-capacity relays are not present in the US, which implies that a single relay can add a maximum of 500 Mbit/s to the total required DoS bandwidth for this country. We provide an overview of the

bandwidth distributions for all guards and exits in the top 5 countries in the appendix (cf. Figure 6.8).

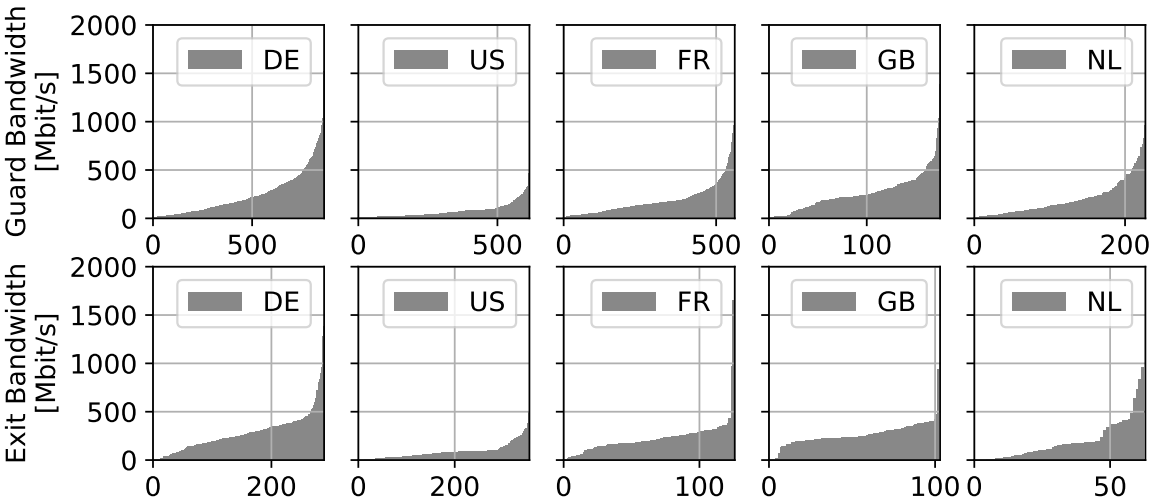

Figure 6.8. Individual relay bandwidths per country separated by Guards (top) and Exits (bottom).

For estimating the practicality of a multi-target denial-of-service attack we consider an adversary who aims to increase their chances for successfully conducting a traffic analysis attack. We consider a nation-state adversary who is able to access all relays located in Germany. Essentially, this scenario resembles a simple statistical calculation. Due to the fraction of 35.7 % of relay bandwidth in DE, the adversary is already in a comfortable situation in being able to access more than one third of exit traffic initially. However, targetedly disabling all exit relays in the other four countries we consider, requires roughly 215 Gbit/s of DoS bandwidth (cf. Table 6.4). In return, 44 % of Tor's overall exit bandwidth is rendered unavailable. Within the set of remaining exit relays, the bandwidth fraction of relays in Germany increases to 64 %, which means that the chances of exit traffic using a relay in the adversarial area have almost doubled.

Cost Estimation The cost to conduct the denial-of-service attack is mainly driven by the cost to produce large amounts of traffic. To estimate the cost, we take into account the amount of traffic that is required to

stress a relay's link bandwidth for one hour. This time-span is sufficient for a targeted attack over a limited period of time. Fully utilizing a link bandwidth of 500 Mbit/s for one hour requires an adversary to generate 225 GB of traffic. This amount seems appropriate for our estimation since, e. g., every relay in the US has a lower assumed link bandwidth. Table 6.5 provides an overview of the corresponding cost using a few large server providers. This means that disabling one relay with an assumed link bandwidth of 500 Mbit for one hour can be purchased for around $ 20. The cost for 215 Gbit/s of DoS bandwidth for disabling all exit relays in US, FR, GB, and NL for one hour (i. e., 96.75 TB of traffic) sum up to $ 8700. These estimations do not consider the cost of running a Tor relay to conduct these attacks from inside the Tor network. However, since the cost for running a standard server instance (which can be used to run the relay) can be kept well below $ 10 per month and one host has a link bandwidth of 10 Gbit/s (among the major server providers, which is sufficient for attacking 20 smaller targets in parallel), these costs seem negligible.

Table 6.5. Traffic cost for the DoS attack.

Provider	**Cost/GB**	**Cost/500M/hour**
Azure	$0.09	$20.25
AWS	$0.15	$33.75
Google Cloud	$0.12	$27.00

6.5.4. Ethics Considerations

During our measurements, we have taken great care to adhere to the principles of ethical research and did our best not to pose any threats to real Tor users or parts of the Tor infrastructure.

In the experiment to determine the cryptographic overhead of the `nTor` handshake times (Section 6.3.1), we ran a publicly accessible relay for general use in Tor. We did not collect any data other than timestamps

and did not harm the anonymity of Tor users connecting to our relay at any time.

In order to validate the DoS mitigation behavior as a means to enforce Guard Rotation (Section 6.3.3), we also ran a publicly accessible Tor relay. In order to minimize its chances for being picked by other clients, we limited the offered bandwidth rate to make it one of the less prominent relays in the consensus. At no point in time, we monitor or interfere with connections from other users.

During the data collection for the exit prediction case study (Section 6.5.1), we established roughly 257,000 circuits but did not actively create payload traffic utilizing the relays involved. In comparison to Tor's daily load, the amount of traffic we created is negligible and did not impair the use of the system.

6.6. Discussion

The threat vectors introduced in this work serve as a stepping stone for follow-up traffic analysis attacks. As they exploit characteristics of core and defensive features within Tor, these threat vectors are hard to counter and cannot simply be removed through an update of Tor. In the following, we discuss alternative directions that can help to limit the success of the stepping-stone attacks of this work.

6.6.1. Impact of DoS Attacks

Because of its voluntarily operated infrastructure, DoS attacks against Tor can be conducted easily. Prior work demonstrates how adversaries can disable critical nodes in the network targetedly [86, 87]. The DoS mitigation features that have been introduced in response aim to recognize and block excessive circuits, connections, and cells. However, we saw that they also *introduce a new threat vector* (cf. Section 6.3.3). Be-

sides this guard rotation, the current DoS mitigation setup further allows continuing DoS attacks against relays from within the Tor infrastructure.

Improving the DoS Mitigation The guard rotation is a reminder of the elaborate design and deployment of new defensive concepts for a live system. Other than the `nTor` handshake procedure, which is a core requirement for the onion encryption of Tor traffic, the DoS mitigation is a defensive mechanism introduced in response to a specific type of attack. While we see that it protects against DoS attacks targeted at clients, its simple concept does not manage to protect relays, nor does it avoid being exploited for other types of attacks against clients. We recommend updating the DoS mitigation in a way that blocks DoS attacks against relays without restricting maintenance traffic (instead of entirely skipping the detection for known IP addresses) and to add a client notification that allows recognizing an exploit of the client DoS mitigation.

6.6.2. Protecting against Exit Prediction

In contrast, the `nTor` handshake is a core mechanism to enable onion encryption, and every circuit build-up depends on it. However, the handshake messages are not obfuscated, and their end-to-end timing allows us to derive the round trip times between single hops of the circuit.

Timing Obfuscation As transmissions through the Internet are often affected by asymmetric routing or congestion, the attack already uses a noisy information source. For a countermeasure, we can use these effects and add further random delays to messages of the handshake. Unlike mixing an entire connection, which introduces an unacceptable overhead, delays in the handshakes are limited to only a few messages and keep the overhead to a minimum. Another option is the use of pluggable transports [20, 124, 236] that obfuscate Tor entry traffic.

Table 6.6 lists the performances of the timing-based exit prediction (AUC of success rates per analyzed exit bandwidth, cf. Table 6.2) for the top 10 countries in terms of exit bandwidth. We compare the performances of the time-based prediction based on original handshake timings and randomly delayed handshake timings. The left column (*No Delay*) corresponds to the *time* column in Table 6.2. The delay amounts in the other columns denote the maximum delay added to each handshake; the individual delay for each handshake was drawn uniformly at random between 0 and the maximum delay.

The results imply that performances of the timing-based prediction can be reduced by up to 21 % (US) when we introduce random delays with a maximum of 0.1 s in the handshakes. This amount of time seems sufficient since we do not observe any added gain when increasing the maximum delay to 0.2 s. With random delays applied, the performance of the timing-based prediction is similar to performances of random and bandwidth-based predictions (cf. Table 6.2).

In conclusion, delaying the `nTor` handshake may be sufficient to prevent from exploiting the timing information at the expense of acceptable timing overhead; however, prediction can still be conducted based on relay bandwidth fractions.

Table 6.6. Effect of timing obfuscation on exit prediction performance.

Country	**No Delay**	**Delay** (0.1 s)	**Delay** (0.2 s)
United States	13.74	10.37 -25 %	10.93 -20 %
Germany	2.86	2.7 -6 %	2.65 -7 %
France	1.48	1.21 -18 %	1.19 -20 %
United Kingdom	2.51	2.26 -10 %	2.79 11 %
Switzerland	3.94	3.62 -8 %	3.47 -12 %
Netherlands	3.95	3.57 -10 %	4.01 2 %
Austria	6.11	5.68 -7 %	5.29 -13 %
Sweden	1.29	1.16 -10 %	0.94 -27 %
Romania	1.58	1.34 -15 %	1.13 -28 %
Canada	1.91	1.74 -9 %	1.7 -11 %

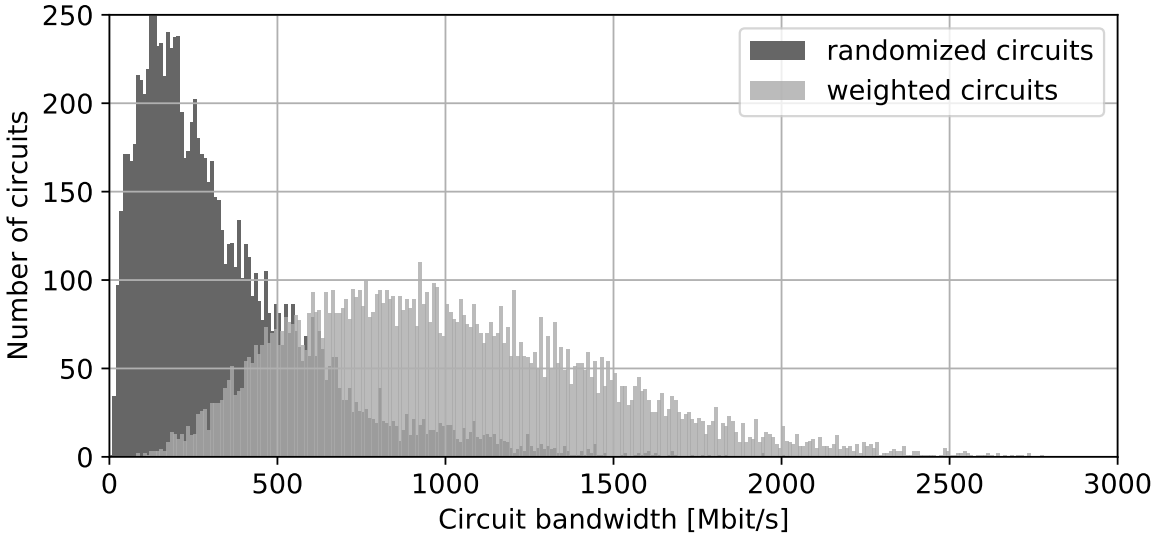

Figure 6.9. Bandwidth comparison for standard Tor circuits vs. circuits with randomly selected relays.

Randomized Relay Selection Adding randomness to the relay selection process may be helpful to hamper adversarial strategies. However, it must be carefully evaluated to what extent the trade-off between security and performance can be further shifted towards security while paying with additional latency.

We compare the circuit bandwidths (i. e., the sum of bandwidths of the three relays in the circuit) of standard Tor circuits and circuits with randomly selected relays. Figure 6.9 shows the distributions of circuit bandwidths for 10,000 circuits of each type. The average circuit bandwidth of 1010 Mbit/s for weighted circuits (median: 955 Mbit/s) decreases to 350 Mbit/s (median: 270 Mbit/s) for random circuits, which comprises an average reduction down to one third.

Besides the connected performance reduction for Tor users, this issue may also cause problems for relay providers. With no adequate load-balancing mechanisms in place (i. e., bandwidth-based selection), especially lower bandwidth relays are used more frequently, even when their original bandwidth capacities are exceeded. Eventually, this renders them unavailable. We leave a more detailed evaluation of these issues an open task for future research.

Uniform Infrastructure The skewed distribution of relays makes it difficult to avoid certain combinations of nodes. For example, avoiding a nation-state adversary in one of the main infrastructure-providing countries leads to severe performance impairments for a user. Prior work addresses such circumvention of untrusted areas and comes to the conclusion that geographical avoidance is possible from a technical perspective but infeasible for specific countries in the Tor infrastructure [98]. A more uniform distribution of nodes, and bandwidth in particular, improves this situation. However, we emphasize that the Tor network consists of a voluntarily operated infrastructure without any central management of relay locations.

6.7. Conclusion

We introduced a set of stepping-stone attacks exploiting previously understudied threat vectors within core mechanisms in Tor. These attacks can facilitate traffic analysis attacks in different ways. The result of an exit prediction hints an adversary towards likely candidates for the exit relay in a Tor circuit. This can either reduce their efforts required for traffic analyses by enabling them to act more targeted, or keep them from attempting an attack that will most likely be unsuccessful. In contrast, active adversary intervention in enforcing the targeted and stealthy guard rotation can provide additional attempts for traffic analyses that turned out to be impossible before. Similarly, targeted stressing sets of Tor relays can be used to enforce routing through adversarial areas, additionally affecting the reliability and stability of the Tor infrastructure.

7

Location Revelation in Instant Messengers

Contents

7.1. Introduction

In recent years, instant messengers have become the de-facto standard for mobile communication. They have transitioned into integral parts of daily lives, with the most prominent messenger, WhatsApp, connecting more than two billion monthly active users world-wide [202]. Messengers are used in a wide range of scenarios, from coordinating homework assignments at school [122] and informal communication among working colleagues [114], to social engagement among elderly people [122], and organizing neighborhood watches [47], thus composing large and heterogeneous sets of contacts in one application per user.

7.1.1. Problem Statement

Whenever a user sends a message in a messenger, the client application displays the current status of the message – from being in transit, processed and forwarded by the messenger server, to delivered to the recipient, and (if enabled) read by the recipient [9], often enabled by small symbols such as checkmarks. This is helpful information for users to track if a message has successfully reached its destination.

However, as we will demonstrate, this feature can also serve as a side channel that allows to learn sensitive information about message recipients, such as revealing information about their current whereabouts. Based on characteristics such as the location of a receiver, delivering a message and returning the respective confirmation takes a specific amount of time. Physical transmissions on the Internet are influenced by the travelled distance, they depend on the network topology, i. e., routing and the hops in-between, and processing by the messaging service. However, this timing side-channel is most likely not expected and surprising for many regular users, and comprises an unintended case of information exposure with undesired potential harm to location privacy.

Deriving sensitive information about someone by sending them a few messages is problematic because it is simple, rather unsuspicious, and hard to mitigate. Users cannot effectively prevent receiving messages from people in their contact list, except for permanently blocking them and, therefore, stopping having mobile conversations with them at all.

7.1.2. Contribution

In our work, we conduct a series of experiments in WhatsApp [238], Signal [147], and Threema [217] to evaluate and demonstrate to what extent we can classify different message receivers and their respective locations based on delivery notification timings of a set of subsequently sent messages. We show that *sending messages* using each of these three messengers to *receivers at different locations* results in different and *distinguishable delivery notification timing patterns*.

This issue is critical for multiple reasons: First, all three messengers we examine are generally considered secure as they use end-to-end encryption between clients. It is not intuitive for users that the mere usage of the messenger service may leak information about their whereabouts. Second, Signal and Threema are best known for their focus on privacy – Signal's protocol serves as the blueprint for provably secure key establishment between clients [42] and has been adapted by other applications such as WhatsApp. Leaking information of the user's location contradicts this notion of privacy. Third, a user cannot do much about someone in their contact list sending them instant messages. Other than read receipt that can be turned off by the receiver for privacy reasons, there is no such option for delivery notifications [237].

In order to experimentally validate this concept we need to take into account the server infrastructures of messengers. This information is not publicly shared and it is a challenge in itself to reliably extract the relevant information such as the number and locations of messenger servers. To this end, we conduct experiments to collect and aggregate informa-

tion about the geographical distribution of servers of popular instant messaging services and analyze if and how knowledge about the messaging server in use affects the outcome of the delivery timing evaluation. We note that the server infrastructure setup does not change frequently, so this step would not have to be redone for each user localization attempt. Beyond the proof-of-concept attack done in this work, knowledge about the messenger infrastructure may turn out to be useful for other purposes. In summary, our work provides the following contributions:

1. **Messenger Infrastructure Analysis.** We aggregate and provide an overview of the geographical distribution of servers of mobile messaging services from a series of experiments to discover and analyze their infrastructures.
2. **Empirical Messaging Experiments.** We conduct large-scale measurements collecting the transmission timings of message delivery timings between devices in multiple locations in Europe and the Middle East.
3. **Attack and Countermeasure Evaluation.** We demonstrate to what extent we can distinguish different receivers and their respective locations from each other based on the measured delivery notification timings. We also show that this threat can be mitigated by randomly delaying delivery notifications in the range of a few seconds.

Experimental Overview. Figure 7.1 provides an overview of our experiments for each of the three parts, their results and connections with each other. We start with infrastructure discovery experiments that result in sets of server locations used to determine the infrastructure overhead in the messaging experiments. At the core of our study, we use sequences of message delivery notification timings to classify receiver locations at different granularity levels and measure the accuracy.

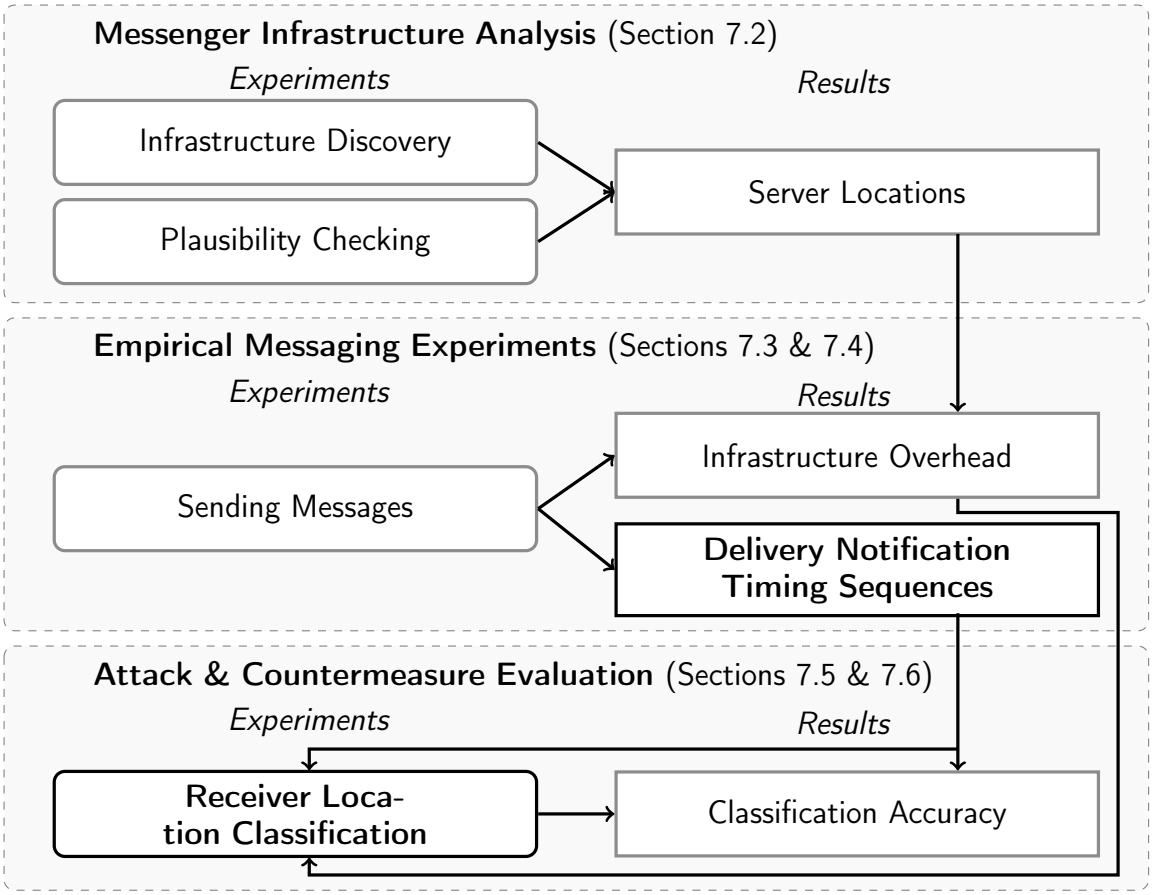

Figure 7.1. Structural overview of the sequence of experiments (rounded nodes) and their outcomes (square nodes) in this work and how the three main parts build upon each other. The main part of our evaluation is highlighted in bold.

Disclosure Process The timing side channel we unveiled in this work may potentially affect the location privacy of millions of messenger users. Following the guidelines of responsible disclosure, we got in contact with the providers of the messenger apps (Signal, Threema, WhatsApp) and reported the vulnerability in May 2022. Whereas Signal and WhatsApp have not acknowledged the issue as of October 2022, we have exchanged ideas for mitigating the problem with Threema and they are currently evaluating how specific countermeasures (cf. Section 7.6) would affect user experience.

7.2. Messenger Infrastructure Analysis

Our first goal is to obtain a comprehensive overview of the infrastructures of the messengers we use in our experiments, i. e., for Signal, Threema, and WhatsApp. For the delivery notification timing analysis, knowledge about the infrastructure is crucial to assess the different parts of the connection between sender and receiver, their distances, and timings.

7.2.1. Discovery and Aggregation

In order to gain first insights into the messenger infrastructures, we conduct a set of experiments to identify servers used by messaging services. In the first step, we set up two smartphones running client applications for all messengers under consideration and capture their network traffic when the applications are running. From the collected captures, we extract the IP addresses of the servers that the application on the smartphone connects to. Since we assume that messenger servers are geographically distributed, the resulting sets of IP addresses may only represent specific fractions of the messenger infrastructures, i. e., they comprise servers near to our own location.

To broaden the perspective derived from our local observations, we perform a two-step DNS analysis, as follows:

(1) For all IP addresses that appear in the communication using one of the messaging applications, we perform reverse DNS lookups to learn what (sub)domain names are used by the messenger operations.
(2) For each domain name in the set derived from reverse look-ups, we perform federated DNS resolving from multiple locations across all continents.

We continue to describe the exact procedures for each messenger individually.

Signal

For Signal, two specific IPv4 addresses are in use. Reverse DNS lookups point to the same domain name operated by Amazon Web Services (AWS), also when we perform these lookups from different geographical locations. When we resolve the resulting domain name, the same two IP addresses are returned, irrespective of the location. Even though the order of the two addresses varies, there is no indication that one address is preferred over the other at specific locations.

Threema

For Threema, we identify two similar IP addresses from the same IPv4/24 address range, for one of which the reverse DNS lookup points to a `threema.ch` domain name. Reverse lookup fails for the other address. We manually identify several more IP addresses whose domain names are resolved to `threema.ch`, resulting in an extended set of 12 IP addresses. However, it is unclear if all these IP addresses are actually used for the messaging application of if they serve other purposes related to the same domain.

WhatsApp

Our reverse domain name resolving of server IP addresses reveals that WhatsApp establishes connections to servers in five different domain name ranges. Additionally, different servers within the same domain name range have been used. Irrespective of the location at which we perform the reverse DNS lookup for a particular IP address, it is resolved to the exact same domain name. Across the three messengers, we discover the largest number of different IP addresses when we explore the network traffic of WhatsApp.

The WhatsApp domain names within the same namespace only differ in 3-letter strings which appear to be IATA airport codes* near our experimental locations. Random checks of additional domain names with the identifier replaced with different ones (in other regions all over the world) reveal further IP addresses, strengthening our assumption.

Since all tested domain names resolve to similar IP addresses in five different IPv4/16 subnets, we conduct a full search of the respective address ranges. We record all domain names and their corresponding IPv4 addresses that contain a reference to WhatsApp (cf. Table 7.1). We further extend the resulting set by manually spot-checking even more identifiers, which leads to a small number of additional servers. In total, our set of discovered WhatsApp servers comprises 410 server instances using 143 different location identifiers.

Table 7.1. Namespace prefixes used by WhatsApp servers.

Namespace (Prefix)	**Number of IPs**	**Number of Locations**
fna-whatsapp	126	75
whatsapp-chatd-edge	94	73
whatsapp-chatd-msgr-edge	92	72
whatsapp-cdn	92	72
whatsapp-pp	6	4
Total Unique IPs/Locations	410	143

7.2.2. Location Analysis

In the next step, we map messenger servers to their individual geographical location and validate the mapping with the help of simple plausibility checks. We initially map each messenger server identified in Section 7.2.1, i. e., their IP addresses, to a specific geographical location. We use dif-

*`https://www.iata.org/en/publications/directories/code-search/?airport.search=`

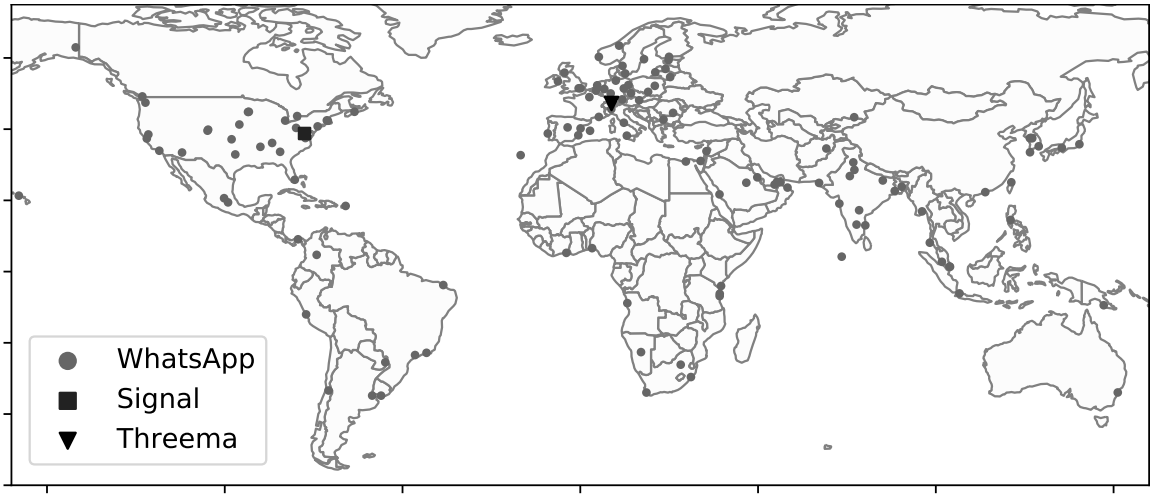

Figure 7.2. Locations of Signal, Threema, and WhatsApp servers around the world (larger version in the Appendix).

ferent strategies depending on the information that we can obtain per messenger.

Little official information about messenger infrastructures is made public by their providers. In the set of messengers we explored, only Threema mentions that their servers are located in the Zurich area, Switzerland [218]. For Signal, no official information is available but several sources indicate that servers are hosted by AWS at the US east coast [14,53,183,240] which is presumably located near Ashburn, VA.

The only information we find with relation to WhatsApp is a list of the locations of Facebook data centers on their website [61]. It is, however, unclear if these locations are also related to WhatsApp. We additionally take into account the presumable IATA location identifiers within the domain names associated with IP addresses used by WhatsApp. We perform look-ups for all 143 codes that appear in our data set and use the resulting city as baseline location for the server. In a few cases, identifiers could not be resolved – and we manually annotate them. For example, the codes *frx* and *frt* most likely belong to the area of Frankfurt, Germany (whose original IATA identifier is *fra*).

We continue with a series of systematic Ping and Traceroute experiments from different geographical locations using a public API provided

by CheckHost [37]. Over a period of four weeks we collect ping and routing information to all messenger servers. To confirm a location candidate as correct, we require that the shortest Ping time is received by the probe host that is closest to the location candidate and only accept minor deviations.

Whereas for WhatsApp and Threema the results are consistent and confirm our initial assumptions about the baseline, the case is more difficult for Signal. Ping information is heavily inconsistent with results being within less than 10 ms from all different continents, which suggests that they are returned from different physical locations close to each of the probing hosts. While Traceroute information can only be partially retrieved for Signal, they include traces with hosts that are likely located in the US, which again strengthens the initial assumption of Signal servers to be US-based.

Figure 7.2 shows our extracted geographical overview of the server locations for the three messengers.

7.3. Message Status Timing Side Channel

The main idea of the attack we present is the use of a timing side channel provided by message status information to derive characteristics of a target user's Internet connection. Whenever two users are in each other's contact list of a mobile messaging application, i. e., they have accepted to be in a conversation on that messenger, the application shows status information for exchanged messages.

Small icons (e. g., check marks) along with each message indicate whether a message has been sent to the messenger server, delivered to the receiver, or read by the receiver. The messages between users as well as the information about the message status are exchanged through TCP messages between the client application and the messenger server. We measure the time between sending a message (i. e., the TCP packets containing the message leaving the sender's device) and the server and delivery confir-

mations (i. e., the TCP packets containing these confirmations) arriving at the sender's device. Observing the resulting timing difference allows us to reason about characteristics of the receiver, such as their location, or their network connection. A schematic overview of the information flow is depicted in Figure 7.3.

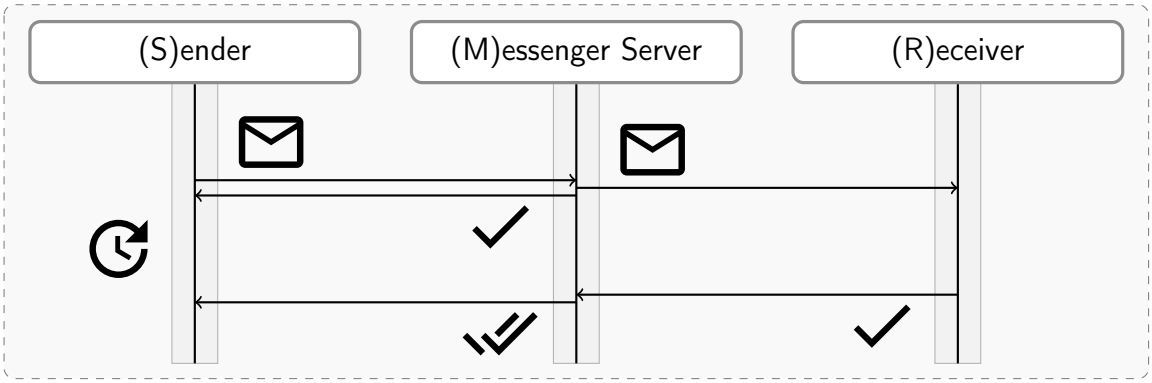

Figure 7.3. Schematic overview of the message flow from the perspective of the sender. The illustration is simplified since sender and receiver can be connected to different messenger servers.

Factors such as the travelled distance between sender, server, and receiver, routing through the Internet between these entities, as well as processing at the messenger server and at hops in-between can affect the observed timings. Repeatedly sending messages to receivers under different conditions (e. g., location, network connection) and observing the timings between messages allows us to learn characteristics of the timings under these conditions in a controlled setup. For different receiver locations, the duration or the distribution of RTTs may be different, e. g., longer times likely represent longer distances between the communication partners.

Within our experiments, we demonstrate to what extent it is feasible to determine certain receiver characteristics upon observing delivery notification timings.

7.3.1. Threat Model

From a *technical* perspective, the adversary is required to operate a regular smartphone that is capable of running a messenger application. The adversary additionally needs to be able to access and analyze their own TCP traffic to extract timing information. This traffic can be captured either on a node in their local network, or directly running on the smartphone when running a packet capture app.

As an *operational* requirement for the attack, adversary and victim must be in each other's contact lists in the messenger. Thus, the threat is limited to parties who likely know each other, as the attack can only be conducted against users who have added the adversary to their contacts. However, the various contexts in which people have messenger conversations, be it in personal (extended family, acquaintances), professional (e. g., work collaborators) or other contexts (e. g., interaction with public institutions, clubs, authorities, within neighborhoods) in combination with low technical requirements still yield a considerable threat scope within social circles, e. g., for stalking.

In an initial training phase, the adversary sends messages to the victim and learns timing characteristics while knowing their whereabouts. Subsequently, the adversary can send new messages to the victim, and determine their location or network connection out of the set of previously seen plausible ones. Since the attack entails sending messages, the adversary's behavior might be observed by the victim and appear suspicious. Therefore, the attacker might leverage timings of messages they send anyway which would, however, narrow down the practical threat scope to people who regularly exchange larger numbers of messages.

7.3.2. Setup

We conduct measurements while sending messages between multiple smartphones in different geographical locations. Our setup comprises two types of devices:

(i) *Active* devices are used to send messages to other devices. Each active device is connected via USB to a computer scheduling the experiment and controlling the smartphone via Android Debug Bridge (ADB).

(ii) *Passive* devices are used to receive messages from active devices. The only requirement for a passive device is having an active Internet connection.

We conduct two rounds of measurements serving different purposes:

1) In the first round, we conduct long-distance measurements with devices distributed across different countries. During this round of measurements, each device is assigned a specific, permanent location. Out of three devices for active measurements, two are located in Germany (*DE-11* and *DE-12*) and one in Greece (*GR-11*). Our setup comprises three more passive devices, located in Germany (*DE-13*), the Netherlands (*NL-11*) and the Middle East (*AE-11*). This experiment is meant to demonstrate a proof of concept that the message-status timing side channel actually exists. For the sake of simplicity, all devices operated on a WiFi Internet connection for these measurements.

2) In a second round of measurements, we send messages from a single active device to passive devices at locations closer to each other, i. e., within the same city, and also rotate passive devices through these locations. Furthermore, passive devices switch between WiFi and cellular Internet connections. We replicate this type of setup in Germany (*DE-2X*) and the Middle East (*AE-2X*). This round of measurements is meant to demonstrate a more practical and realistic attack scenario, imitating a natural everyday behavior of a target messenger client, e. g., being at their home and work location (WiFi) and moving in between and around (cellular). Furthermore, this second round also shows to what extent the attack works at a smaller scale, which is less obvious than comparing timings at country level.

Table 7.2. Devices and locations in our measurements.

ID	Model (Year)	Type	Locations
Round 1			
AE-11	Huawei P40 (2020)	P	AE-A (W)
DE-11	Xiaomi Mi A3 (2019)	A,P	DE-A (W)
DE-12	Huawei P8 Lite (2017)	A,P	DE-B (W)
DE-13	OnePlus 7 Pro (2019)	P	DE-B (W)
GR-11	Samsung Note 10+ (2019)	A,P	GR-A (W)
NL-11	Samsung S6 (2015)	P	NL-A (W)
Round 2 (United Arab Emirates)			
AE-21	Huawei P40 (2020)	A	AE-B (W)
AE-22	Samsung Note 10 (2019)	P	AE-A, AE-D (W, 4G+)
AE-23	Samsung S22 (2022)	P	AE-B (W, 5G)
AE-24	Nokia X10 (2021)	P	AE-C (W, 4G+)
Round 2 (Germany)			
DE-21	Huawei P8 Lite (2017)	A	DE-A (W)
DE-22	Huawei P8 Lite (2017)	P	DE-A (W), DE-B (W, 4G), DE-C (W)
DE-23	Google Pixel 3a (2019)	P	DE-A (W, 4G), DE-B (W), DE-D (W)
DE-24	Samsung S6 (2015)	P	DE-A (W), DE-B (W, 4G), DE-E (W)

Locations: *AE-A,B,C,D:* Abu Dhabi, UAE; *DE-A,B,D,E:* Bochum, Germany; *DE-C:* Essen, Germany; *NL-A:* Nijmegen, Netherlands; *GR-A:* Athens, Greece

In Table 7.2, we provide an overview of the devices and their locations involved in the two rounds of our experiments. For each location, we also indicate whether we use WiFi (W), or cellular (4G/4G+/5G) connections, or both for measurements at the respective location. Additionally, Table 7.3 lists distances between locations for all three setups.

7.3.3. Measurement Procedure

We measure the time it takes for a message from a sender device to be delivered to the messenger server and to the recipient. To this end, we capture an active smartphone's network traffic directly on the device using the *tPacketCapture* app. The phone is connected to a computer via USB and a Python script controlling the phone via *Android Debug Bridge (ADB)* automatically schedules the processes of sending messages

Table 7.3. Distances [km] between device locations.

Round 1					*Round 2 (UAE)*				*Round 2 (Germany)*				
	DE-B	NL-A	GR-A	AE-A		AE-B	AE-C	AE-D		DE-B	DE-C	DE-D	DE-E
DE-A	1.5	98.7	1972.9	4981.0	AE-A	7.8	0.4	19.3	DE-A	1.5	14.4	3.4	5.4
DE-B		97.5	1974.4	4982.2	AE-B		8.1	24.9	DE-B		13.5	2.3	4.0
NL-A			2065.8	5079.5	AE-C			18.9	DE-C			11.2	10.3
GR-A				3263.3					DE-D				2.3

and capturing network traffic. The script uses system commands to open and close the packet capture and messaging apps, and interacts with the UI to navigate within the apps, i. e., simulates human touch input to select contacts or type messages.

In a single experiment iteration, the phone subsequently sends a series of five messages to all receivers, with each messenger that is running on the sender and on the receiver device. The texts of the messages remain the same throughout the whole experiments. The first four messages are short and only consist of a single wordeach, while the last message is a whole text paragraph. We send the first four messages at an interval of 10 seconds to allow for the confirmations to arrive before sending the next message, while we increase the waiting time before the last messages to 20 seconds in order to accommodate the longer time it takes to type the long text, thus facilitating the analysis of the packet captures. The measurement procedure is complete when all iterations have terminated successfully for all recipients and their corresponding messaging applications. Algorithm 1 shows our measurement procedure.

We repeat this procedure over a period of several weeks in July and August 2021 for Round 1 and March to April 2022 for Round 2. Whereas the physical locations of receiving devices remain unchanged throughout the Round 1 measurements, we collect data for at least one week for each location a receiving device was placed at in Round 2. In total, we use more than 240,000 messages sent during the two rounds of experiments for evaluation.

Algorithm 1: Texting Thumb

input : A list of *messengers* which are supported applications of the receivers
input : A list of *receivers* according to the contact list
input : A list of *words* which are sent to the receivers consecutively
output: *void* function

```
1  sleep_time = 10;
2  num_of_messages = 5;
3  for receiver in receivers :
4      for messenger in messengers :
5          start_pcap ();
6          start_app (messenger);
7          open_chat (receiver);
8          for i ← 0 to num_of_messages - 2 :
9              send (words[i]);
10             sleep (sleep_time);
11         sleep (sleep_time);
           /* Send the long text                    */
12         send (words[num_of_messages - 1]);
13         close_app (messenger);
14         stop_pcap ();
```

7.4. Descriptive Dataset Analysis

Using the setup described in Section 7.3.3, we collected our dataset and use it in the further investigations.

7.4.1. Data Processing

For each measurement iteration, we evaluate the recorded packet captures to determine the elapsed time between a message sent by the sender and the notifications (by the server and receiver) that return to the sender.

Since the messengers we consider use multiple layers of encryption (i. e., end-to-end encryption between the communication partners and TLS-

encryption for connections between clients and servers on the transport layer), we are not able to access the contents of the communication. Yet to analyze the communication flow and identify the messages and confirmations, we develop heuristics from sample captures. We analyze characteristics of the network traffic such as packet sizes, their order and flow direction, which is a common technique, e. g., for traffic analysis [38, 209].

Within our analysis, we only consider traffic between the sender device and IP addresses within the IP address range(s) of the respective messaging service (cf. Section 7.2). We are interested in sequences of packets of the form as illustrated in the information flow overview in Figure 7.3. The message sent by the sender usually consists of one or more outgoing TCP packets whose destination is one of the messenger servers. After a message has been sent, there is one incoming TCP packet containing the server notification, coming from the messenger server. Finally, once the receiver has confirmed that they have retrieved the message, there is another incoming TCP packet containing the delivery notification. From the sender's perspective, this packet is also coming from the messenger server. These observations are based on a first manual visual inspection of a small set of packet capture files.

Taking into account the aforementioned network traffic structure, we conduct our detailed packet capture analysis in two steps:

(1) Identifying typical packet sizes of server and receiver notifications.
(2) Matching sequences of TCP packets to determine round-trip times between sending a message and receiving the notifications.

Identifying Packet Sizes of Notifications

In the first step, we use a subset of $n = 1000$ randomly selected packet capture files and analyze the packet sizes of the two types of incoming packets (i. e., the notifications from server and receiver). To make sure that we only consider packets that contain these notifications, we limit

Table 7.4. TCP packet lengths of notifications.

Messenger	Bytes (Server)	Bytes (Receiver)
Signal	123–124	773–828
Threema	38	158–390
WhatsApp	68–69	61–62

our first analysis to sequences of packets that appear right after one another and right after the message has been sent.

We then analyze the lengths of the two inbound packets in all matched packet sequences across all packet capture files to identify the lengths of the packets containing the two types of notifications. We evaluate the frequencies of packet lengths, conducting an additional round of manual plausibility checks within the traces. The results are listed in Table 7.4. Most notably, the length of the packet containing the notification that a message has been delivered to its receiver in Threema is uniformly distributed between 158 and 390 bytes. In contrast, the other notifications have smaller variations in packet length: Signal's notifications range from 773 to 828, and WhatsApp's from 61 to 62.

Matching Packet Sequences to Determine RTTs

In the second step, we systematically analyze all packet captures we have collected during the two rounds of measurements. Since we now know the sizes of packets we are interested in, we omit the requirement of packets to appear right after one another in the correct order. This helps us to also identify messages whose delivery notification is delayed, or when the traffic patterns we are interested in interferes with other packets exchanged between the client application and the messenger server. We first identify the two inbound packets (i. e., the two notifications n_1 and n_2) based on their size and match them with the latest outbound packet (i. e., the message m) sent before those two packets arrived. An example is illustrated in Figure 7.4.

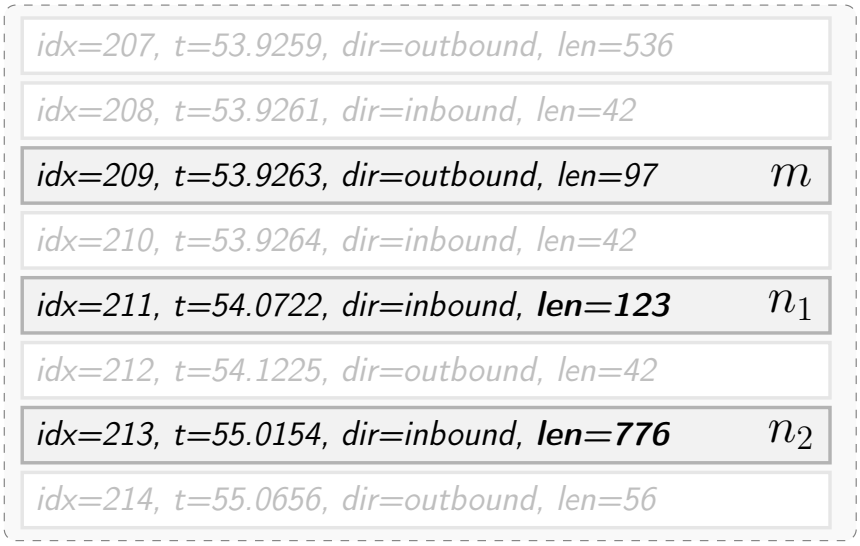

Figure 7.4. Excerpt from an example packet capture with the three identified packets of interest highlighted.

We use the timestamps of the three packets (i. e., $t(m)$ for message m) to determine the notification round-trip times (RTT) between (S)ender and (M)essenger Server, and (S)ender and (R)eceiver:

$$\begin{aligned} RTT_{S,M} &= t(n_1) - t(m) \\ RTT_{S,R} &= t(n_2) - t(m) \end{aligned} \tag{7.1}$$

Finally, we calculate the hypothetical RTT between (M)essenger Server and (R)eceiver:

$$RTT_{M,R} = RTT_{S,R} - RTT_{S,M}. \tag{7.2}$$

Additional Notes on Signal in the UAE

In the Signal data collected in Round 2 in the UAE, we observed different traffic characteristics. In particular, there is only one specific packet returned from the server – presumably containing both confirmations from server and receiver. Thus, we cannot determine the difference between the two but only consider $RTT_{S,R}$ for our analysis.

7.4.2. Delivery Notification Timings

We now present a first view into our delivery notification timing dataset. We start by analyzing the measured times in relation to the traveled distance, and later continue with distributions of timings to different receivers.

Timings and Distances

We are first interested in the relation between the timings we observe and the traveled distances between sender, messenger server, and receiver. To this end, we analyze what messenger servers have been picked on the sender's side and leverage the findings from our messenger infrastructure analysis (cf. Section 7.2) to determine the distances from the server to sender ($dist_{GCD}(S, M)$) and receiver ($dist_{GCD}(M, R)$), respectively. We emphasize that the receiving device might be connected to a different server (location) than the sender – however, from the attacker's position (i. e., the sender), this information cannot be further resolved. We can then analyze the relation between timings and distances for the two segments.

In Figure 7.5a, we see a slight tendency for minimum timings to increase for longer distances between sender and server (for Threema and Whatsapp), even though timings are largely scattered for similar distances. In Figure 7.5b, there is, again, a comparably small set of distances between servers and receivers, and timings being scattered a lot without clear trends. Since our experiments only cover a small set of distances between devices, and only consider Great Circle Distances (GCD) between entities, without taking into account the actual routing through the Internet topology, our dataset does not allow to develop a generalized model to put timings in direct relation to the traveled distances. To reduce the noise introduced into our data at this stage, we continue with focusing on the timings between messenger server and receiver, i. e., we use $RTT_{M,R}$ in subsequent analyses.

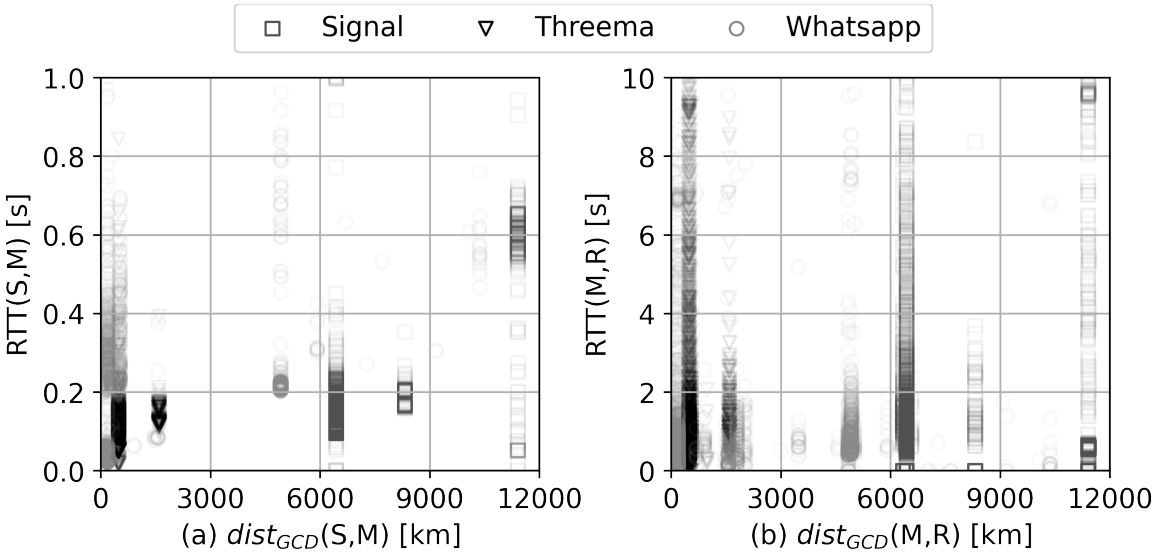

Figure 7.5. Round trip time distributions of distance splits for (a) sender to server and (b) server to receiver – (each with 2000 randomly sampled timings per measurement round). Y-axes have different ranges since the time it takes to return the confirmation from the receiver is considerably longer.

Differences between Receiver Characteristics

In the next step, we analyze to what extent timings we collected comprise differences between receivers, or their characteristics, respectively.

We first compare the measured $RTT_{M,R}$ between receivers the different countries involved in the first round of experiments. Figure 7.6 illustrates distributions of these timings of messages sent from device *DE-11* to receivers in different countries for each messenger. For all messengers, we observe that timings to Germany are shorter (lower medians) and tighter distributed (smaller boxes). Shorter timings for Germany are the result we expect in this case, since all messages have also been sent from a device in Germany. Whereas the differences between the medians are smaller for the other countries, distributions have different widths (heights of boxes) or are differently shifted (position of boxes).

While differences between the distributions of notification timings to receivers in different countries can be easily identified, we also analyze if such differences also exist at smaller scale. Moreover, we cannot en-

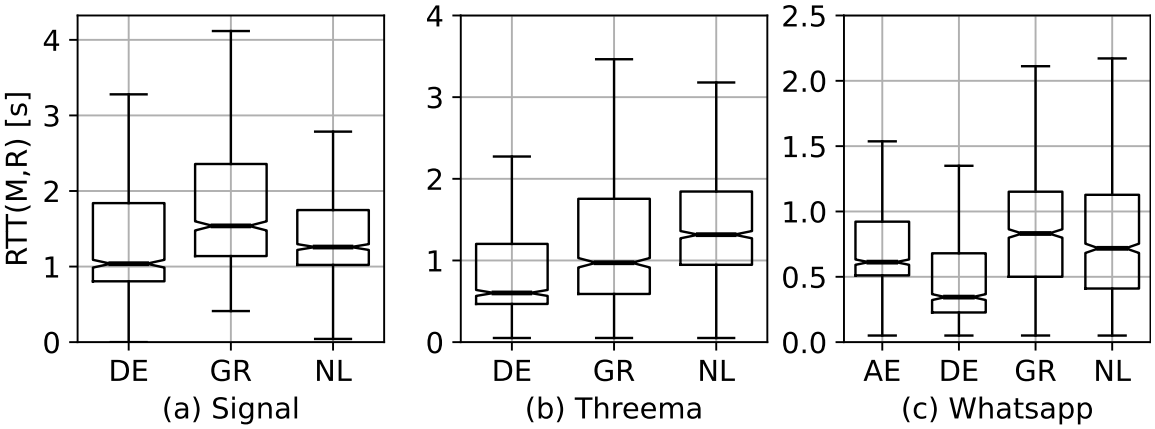

Figure 7.6. Messages sent from device *DE-11* to receivers in different countries. Y-axes have different ranges since we only intend to highlight differences within each messenger.

tirely exclude that these differences are partially grounded in the devices itself, since in the first round of measurements, each country location corresponds to a different device. In this regard, we now compare notification timings of messages sent to device DE-22 at its different locations in Germany during the second round of measurements.

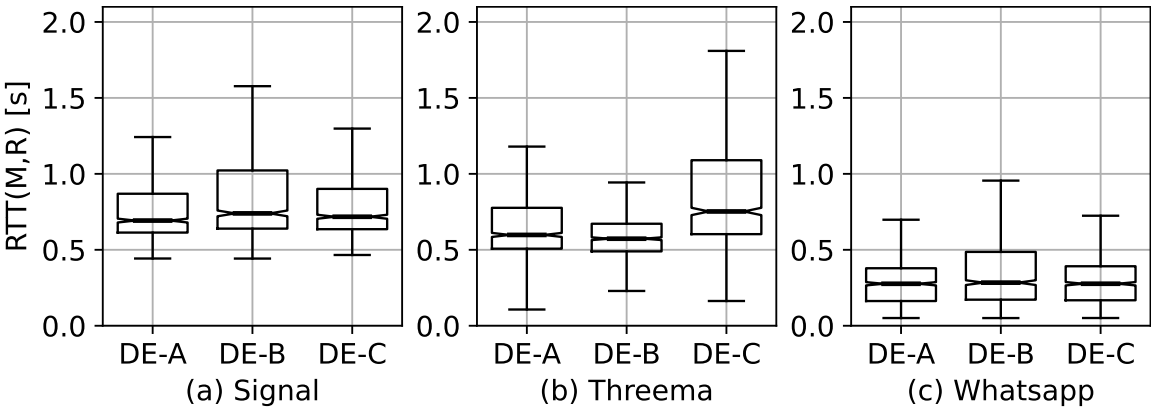

Figure 7.7. Messages sent to device *DE-22* separated by the device's location.

Figure 7.7 shows the distributions of timings to the three locations. Differences appear to be much smaller than those on the per-country level, we can only observe small variations in, e. g., medians or ranges of timing distributions, indicated by ranges and shapes of boxes.

In the last step, we also compare notification timings sent to the same device depending on its network connection. In this case, differences appear to be larger again, with distributions of timings of messages received over cellular data showing a higher variance (larger box) and being slightly slower, indicated by a higher median (cf. Figure 7.8).

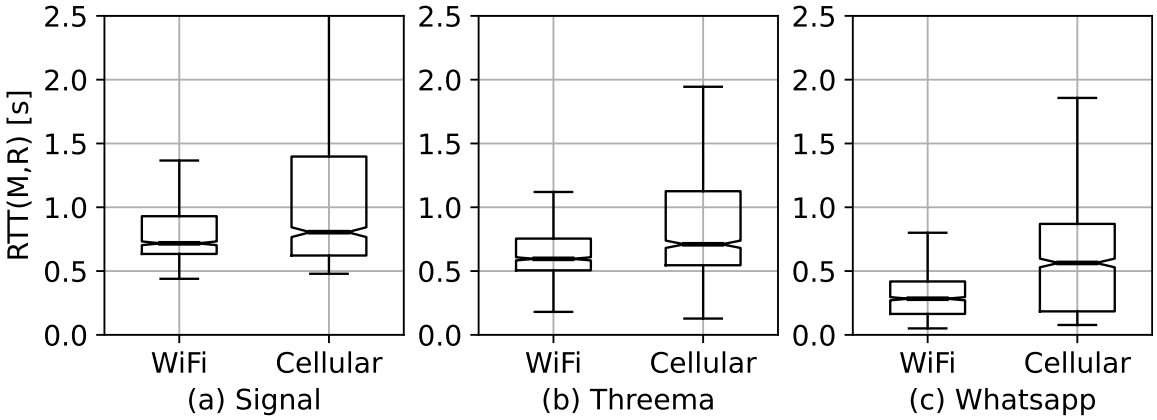

Figure 7.8. Messages sent to device *DE-22* separated by its network connection.

7.5. Delivery Notification Timing Classification

Classifying the timing measurements collected in the experiments can help to determine certain characteristics of the receiver of a message, such as their location. We demonstrate at what scale it is feasible to classify different targets based on delivery notification timing measurements and to distinguish these characteristics from each other.

7.5.1. Classification Tasks

To evaluate and demonstrate at what scale the classification of receivers and their characteristics works, we specify a set of classification tasks at different granularity levels as follows:

(1) *Country*: We distinguish our measurements by the country a receiving device is located in (out of the set of countries we have measurements for).
(2) *Within Country*: We only distinguish whether or not a receiving device is located within a specific country.
(3) *City Location*: We distinguish timings to different locations within the same city. We repeat this classification task for devices individually and conjointly.
(4) *Network Connection*: We distinguish whether a device is using a WiFi or a cellular Internet connection.

According to the designs of our measurement setups (cf. Section 7.3.2), we use data from the first round of measurements for classification tasks (1) and (2), whereas classification tasks (3) and (4) are based on data from the second round of measurements.

7.5.2. Classification Setup and Parameter Tuning

We use *sequences* of delivery notification timings for classification. A sequence is a set of notification timings derived from up to five subsequently sent messages (cf. Section 7.3.3). We repeat the classification with different sequence lengths, starting with $n = 1$, i. e., a single notification round-trip time from a single message.

For each classification task, we analyze the measurement data for each sender device and for each messenger independently. We randomly sample k notification timing sequences from each class, whereas k is the number of timing sequences of the class with the lowest number of sequences. This way, we reach an evenly weighted set of samples per class.

We use convolutional neural networks (CNN) as classifiers, train them with sequences of delivery notification timings from different classes and then measure their accuracy in predicting newly observed timing sequences. This selection is grounded in our own preliminary parameter tuning evaluation and builds upon findings by Rimmer et al. [170], who

extensively evaluate the performance of different types of neural networks for a similar network traffic analysis task (Website Fingerprinting) and report CNNs to perform best when compared to Long Short-Term Memory (LSTM) networks and Stacked Denoising Autoencoder (SDAE) networks. Before we start the actual classifications, we repeatedly run the first classification task with varying parameters to find the optimal classification setup for each of the three types of neural networks, i. e., CNN, LSTM, and SDAE and compare the results. We provide an overview of the parameter tuning configurations in Table 7.5.

Table 7.5. Parameter tuning configurations with best performing settings highlighted in bold.

CNN	**LSTM**	**SDAE**
Activation function tanh, **relu**	tanh, Sigmoid, **relu**	**tanh**, Sigmoid, relu
Optimizer SGD, **Adam**, RMSProp	SGD, **Adam**, RMSProp	SGD, Adam, **RMSProp**
Dropout rate 0, **0.1**, 0.2, 0.3	**0**, 0.1, 0.2, 0.3	**0**, 0.1, 0.2, 0.3
Number of epochs 20, 30, 40, 50, **60**	20, 30, 40, 50, **60**	20, 30, 40, **50**, 60
#CNN input filters 8, 16, **32**, 64	—	—
#Fully-connected layers 1, **2**, 3, 4, 5	—	—
#Neurons on fully-connected layers **50**, 100, 200, 500	—	—
#LSTM layers —	1, 2, **3**, 4, 5	—
#LSTM units —	50, **100**, 200, 500	—
#Encoding layers —	—	**1**, 2, 3

7.5.3. Classification Procedure and Evaluation Metrics

For each classification task, we randomly split the respective data into five portions and use all but one of these portions as training set for the neural network. The remaining portion serves as test set from which all samples are to be classified. For each sample in the test set, the neural network output comprises a softmax result, i. e., assigning each candidate class a probability that the classified sample belongs to this class. Based on the softmax output, we assign each sample the class with the maximum probability, considering this as the classification decision. To avoid model over-fitting, we repeat this procedure until each of the five data portions has served as test set and merge the five classification results, effectively implementing 5-fold cross-validation.

The performance of the classification is determined by the numbers of classifications that identify the correct class (*precision*) and the number of samples in each class that are correctly classified (*recall*). In our evaluation, we focus on precision, i. e., we are interested in the fraction of samples per class that can be correctly identified and how the classifications are distributed for all samples of a particular class. We also analyze changes in classification performance when we vary the sequence length.

We report these detailed results for the first classification task (i. e., distinguishing receiver countries) to provide detailed insights into our evaluation and how it works. For subsequently presented classification tasks, we report overall accuracy results for a large number of different classifications using the maximum delivery notification sequence length (i. e., 5 messages). We do so to provide a broad overview of the varying effectiveness of leveraging the timing side channel in different scenarios. We provide detailed results of all instances of all classification tasks in Appendix C.

Finally, we also analyze the convergence of the classification accuracy depending on the sample size, i. e., we repeat a selected set of classifica-

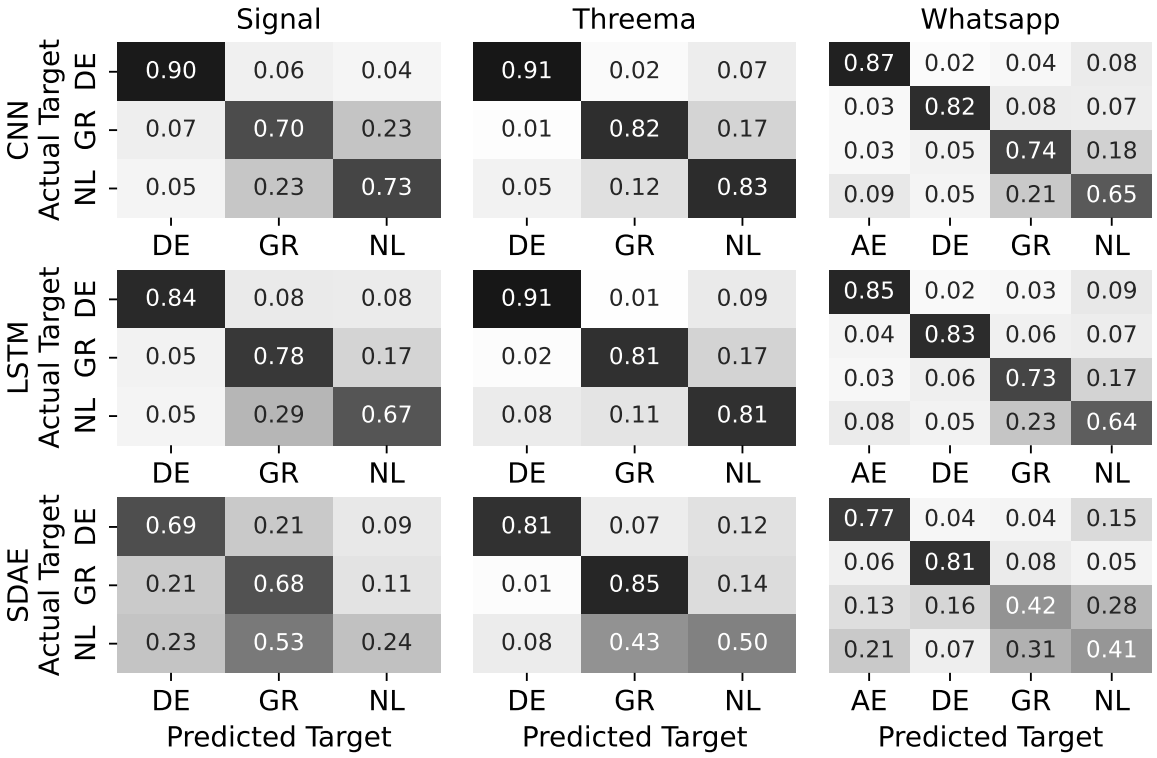

Figure 7.9. Detailed classification results for the receiver *country* based on measurements from sender *DE-11* with three different neural network types. For each classification, numbers report *precision* values, i. e., the fractions of predicted classes (x-axis) given the actual class (y-axis).

tions multiple times with increasing numbers of samples per class, and measure the resulting performance.

7.5.4. Receiver Classification by Country

In the first step, we present the results of the receiver *country* classification for one sender device in Germany (*DE-11*). For WhatsApp, the receiver can be one of four countries (*AE*, *DE*, *GR*, *NL*). For the two other messengers, we cannot present data for *AE* due to the messenger not being available at all in the country (Threema) or too little successful delivery notification measurements (Signal). To this end, we are restricted to the three remaining countries for Signal and Threema.

Detailed results are presented in confusion matrices in Figure 7.9, separated by messenger (columns) and neural network type (rows). The

numbers indicate the fractions of predicted classes for each actual class (*precision* values). A darker principal diagonal in each matrix indicates higher accuracy since numbers on this axis refer to correct predictions. Figure 7.10 illustrates corresponding overall accuracy for this classification tasks for all three messengers depending on the length of the notification timing sequence.

For Signal (left column matrices), the receivers located in Germany can be distinguished from receivers in the two other countries quite well. We observe false classifications mostly between devices in GR and NL. This result is not surprising since timing distributions for GR and NL largely overlap, whereas timings of messages to receivers in DE are lower (cf. Figure 7.6). The overall classification accuracy rises from 60 % for a single timing per sample to 79 % for 5 timings per sample (cf. Figure 7.10) in the case of a CNN classification. For Threema, there is a quite similar outcome. Again, classification works best for receivers in *DE* with most false classifications between *GR* and *NL*. For longer timing sequences, Threema reaches a better overall accuracy of 86 % for 5 timings per sample, compared to 60 % for single-time samples. In the case of WhatsApp, receivers in *DE* and *AE* can be distinguished best from the others and performance increases for longer timing sequences. The overall accuracy is a bit lower for the other two messengers (i. e., 47 % to 77 %).

Regarding the classifier type, CNN and LSTM perform with similar quality with CNN reaching slightly higher performances in most cases. SDAE results are noticeably worse. Therefore, and taking into account previous findings [137, 170], we continue with CNN throughout the remaining evaluations.

Table 7.6 lists precision results per class for the country classification for messages sent with all three messengers from three sender devices (*DE-11*, *DE-12*, and *GR-11*). We also report sample sizes of notification timing sequences there. All results listed in the table refer to the maximum notification timing sequence length, i. e., timings of *five* sub-

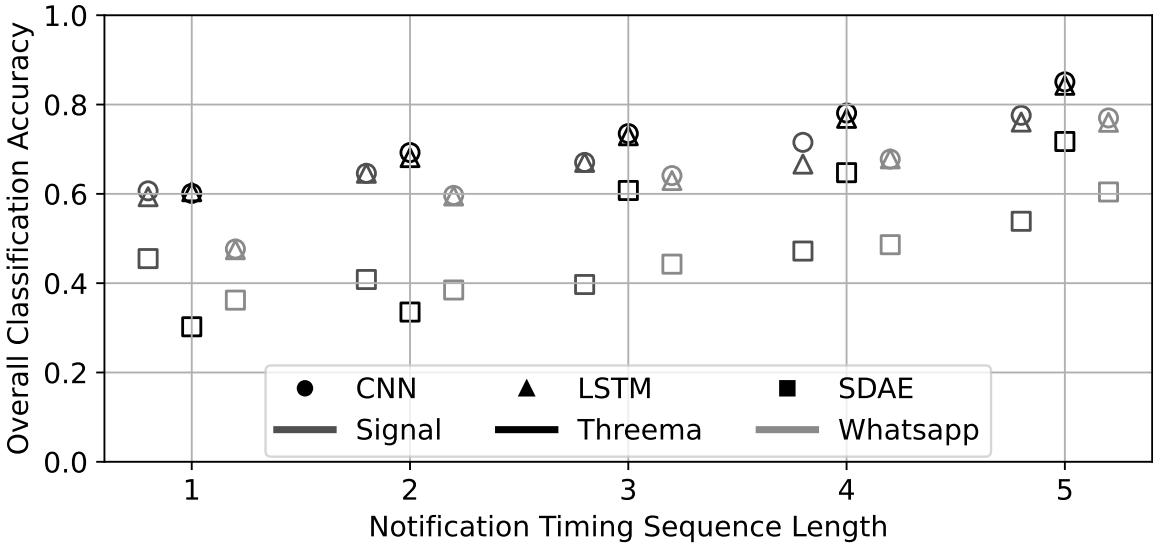

Figure 7.10. Overall classification accuracy (y-axis) for the receiver classification per country, depending on delivery notification timing sequence length (x-axis) and NN type (icon shape)

.

sequently sent messages. The results in the top left block of the table correspond to the numbers presented in Figure 7.9 with each table column corresponding to the principal diagonal axis in the respective confusion matrix.

Country Subsets

We repeat the classification of delivery notification timing sequences with the other devices and for every subset of countries in our data set. The resulting set comprises one more classification of four countries (sender device *DE-12*) and multiple evaluations of all possible pairs and triplets of countries including measurements from all three sender devices. In this context, we only consider the maximum sequence length, i. e., delivery notification timings of $n = 5$ subsequently sent messages.

Figure 7.11 shows the overall accuracy values of the receiver country classification for all combinations of countries in our data set. For smaller target sets, classifications perform better, with overall classification accuracy mostly between 70 % and 90 %. In the case of two countries, some classifications even perform with more than 95 % accuracy. Such

Table 7.6. Detailed precision results for the classification of receiver locations (CNN-based classification).

Sender	**DE-11**			**DE-12**			**GR-11**	
Messenger	SIG	THR	WA	SIG	THR	WA	THR	WA
Classification Task: Country								
AE	–	–	84 %	–	–	94 %	–	95 %
DE	90 %	94 %	81 %	73 %	70 %	77 %	71 %	89 %
GR	77 %	84 %	79 %	53 %	68 %	64 %	–	–
NL	70 %	80 %	63 %	61 %	68 %	53 %	66 %	88 %
Samples/Class	177	527	825	66	60	135	187	168
Overall Accuracy	79 %	86 %	77 %	62 %	69 %	72 %	68 %	90 %
Classification Task: Within Germany								
DE	92 %	91 %	90 %	86 %	84 %	92 %	90 %	90 %
NOT-DE	91 %	94 %	92 %	78 %	85 %	88 %	51 %	94 %
Samples/Class	559	1135	1888	250	180	605	187	349
Overall Accuracy	91 %	92 %	91 %	82 %	85 %	90 %	70 %	92 %

nearly perfect results can only be achieved when timings can be clearly distinguished, which is mostly the case when the candidate locations are far from each other (one receiving device located in the UAE and the other one in a European country). However, also for distinguishing notification timings of messages sent to Germany and to the Netherlands (*DE11-2countries1*), we achieve a classification accuracy of more than 90 % (92 % for Threema and 91 % for Signal and WhatsApp).

Within Country

In the second classification task, we are interested in whether or not a receiver is located in a specific country. Different from the previous task, we are not interested in determining the exact location but only in a binary decision about a specific location. Therefore, we only distinguish notification timing sequences of messages sent to the country we are interested in (e. g., *DE*) from timings to any of the other countries, effectively summarizing timing sequences of all other countries into one

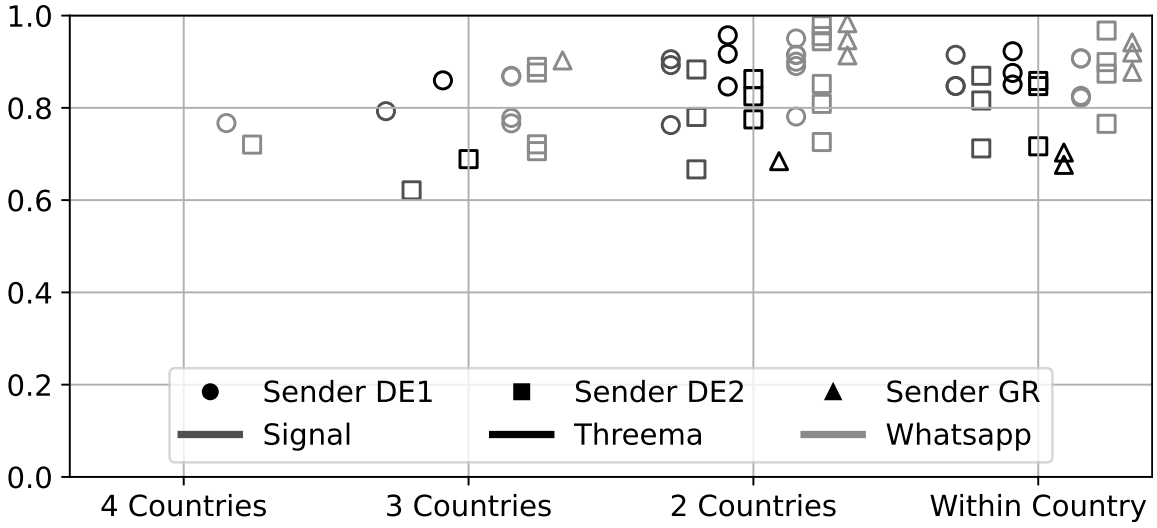

Figure 7.11. Overall accuracy of receiver country classification separately for all possible country subsets for each sender device (icon shape) and messenger (colors).

class (e. g., *NOT-DE*). Technically, this type of prediction is similar to the classification of two countries.

Figure 7.11 also includes accuracy results for all such classifications, with the majority being very similar to the two-country classification. As an example, we provide more detailed precision results for the *Within Germany* classification task in Table 7.6 for all three sender devices.

7.5.5. Receiver Locations Within the Same City

We now present classification results for receivers at different locations within the same city to demonstrate that the timing side channel provided by delivery confirmations also persists at smaller scale. In this case, the end-to-end distances between sender devices, messenger servers, and receiver devices remain roughly the same across all measurements. Similar to the per-country classification, we consider all possible combinations of WiFi locations and subsets and evaluate the classification performance for each of them. Subsequently, we repeat the analysis also including the timing data retrieved from receivers operating on a cellular connection as a separate class. We repeat these analyses for receiver devices indi-

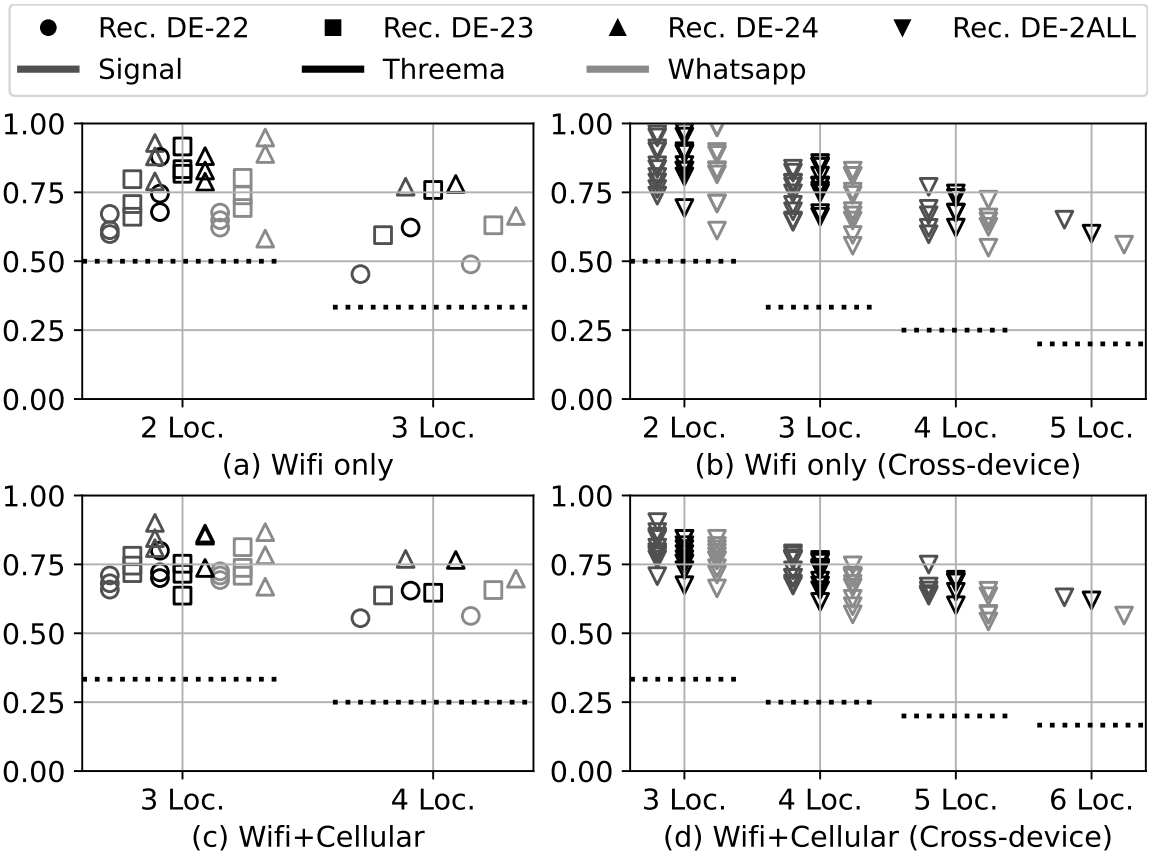

Figure 7.12. Overall accuracy of receiver location classification separately for all possible combinations of locations in Germany. Colors indicate messengers and icon shapes indicate different receiver devices (we refer to Rec. DE-2ALL for the cross-device analysis). Dotted lines indicate the probability of randomly guessing the correct location out of the set of known locations.

vidually and across all devices within the same setup, i. e., the Round 2 measurements in Germany and in the UAE (cf. Table 7.2). Whereas cross-device analyses provide first insights towards the generalizability of receiver location classification models (i. e., whether or not the classification requires training for each individual device), the individual analyses ensure that the classification is not biased by timing artifacts introduced by characteristics of the different devices.

Individual Receivers

The classification results for the three receiving devices in Germany are illustrated in Figure 7.12a+b. The accuracy highly varies between messengers, devices, and the respective combination of locations. Across all combinations of two locations, in each of which the device is connected via WiFi (a), the prediction performance can reach more than 90 % in some cases, e. g., when distinguishing locations *DE-A* and *DE-B* for the receiver device *DE-24* (*2wloc1-DE-24*). On the other side of the spectrum, there are also combinations of two locations which cannot be distinguished at all – a classification accuracy of around 60 % is hardly better than randomly guessing one of the two location candidates, e. g., when distinguishing locations *DE-B* and *DE-C* for device *DE-22* (*2wloc5-DE-22*). For distinguishing three WiFi locations, accuracy is lower with a maximum of 77 % for Signal, 78 % for Threema, and 66 % for WhatsApp (*3wloc3-DE-24*). However, the chance of randomly guessing one location is also lower in this case (33 %).

Identifying the correct location becomes easier when the receiving device operates on a cellular connection in one of them (cf. Figure 7.12b). For distinguishing two WiFi locations and one on mobile data, the classification accuracy is mostly between 60 % and 80 %. Such a scenario could, for example, model home and work locations of the device owner, whereas the cellular connection represents any other place in which the phone is not connected to a WiFi network.

Cross-device Analysis

When distinguishing locations across different devices (cf. Figure 7.12c+d), classification performs similar to the case of individual devices, with accuracy increasing slightly. Such differences might come from individual devices introducing specific timing characteristics into the dataset that facilitate distinguishability of locations.

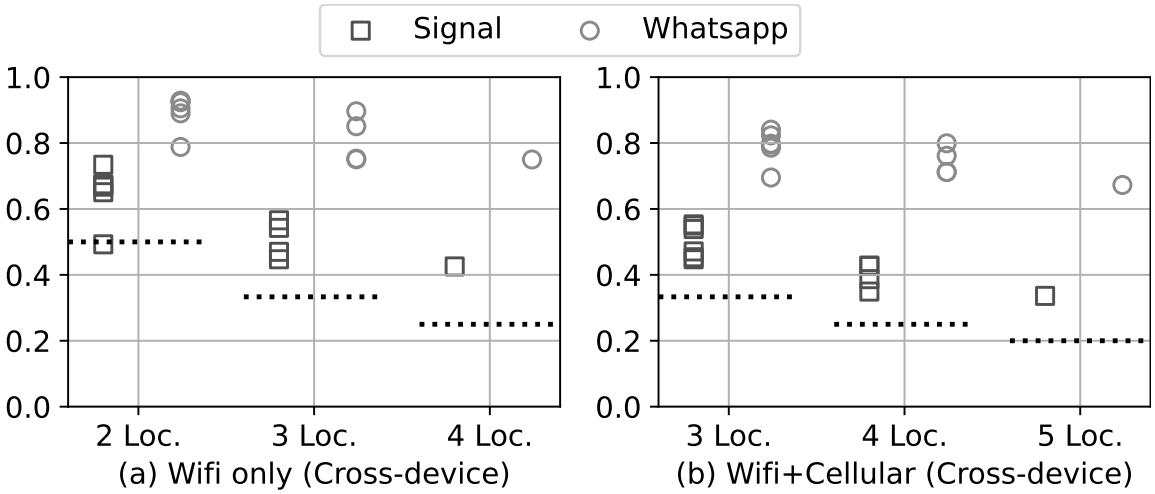

Figure 7.13. Overall accuracy of receiver location classification separately for all possible combinations of locations in the UAE. Colors indicate messengers and dotted lines indicate the probability of randomly guessing the correct location.

For the data collected in the UAE setup, the picture is more diverse. As the results in Figure 7.13 show, both two and three WiFi locations can be distinguished with up to more than 90 % accuracy in WhatsApp, which resembles better performance than comparable classifications in the German setup. However, for Signal, the classification of locations does not seem to work at all, which we attribute to the different structure of message exchange (and in particular the presence of only one confirmation packet) as described in Section 7.4.1.

7.5.6. Receiver Network Connections

Since different locations can apparently be better distinguished when the receiving device operates on a mobile data in one of them, we also analyze if we can generally detect whether a phone is connected via WiFi or using a cellular connection. Being able to distinguish these two cases allows us to determine whether a target is currently in one of their usual locations (i. e., we assume that they are connected to the respective WiFi network there) or not (mobile data).

The results for the evaluation of this classification task are listed in Table 7.7. In the setup in Germany, we can detect the receiver's Internet connection type with high accuracy for all devices for all messengers, both for individual devices and also across different ones. Classifications reach an overall accuracy of 90 % or even above, with only one prediction performing worse (Device *DE-23*, Threema). In the setup in the UAE, predicting the network connection performs on a similar level for WhatsApp. In the case of Signal, results do not seem convincing (50 % corresponds to randomly guessing the connection type), which is in line with results of the WiFi location distinguishability.

7.5.7. Classification Accuracy Convergence

Whereas the results reported for the classification so far always refer to the maximum number of notification timing sequences available for all classes, we are also interested in how many samples are actually required for an accurate classification. To this end, we repeatedly run four specific classifications representing different classification tasks with increasing numbers of notification timing samples. We start with 10 samples per class and increase this number in steps of 10 until we reach 300 or the maximum number of available samples for all classes (if it is lower than 300). Figure 7.14 illustrates the results of these evaluations. We include (a) the receiver country classification based on the

Table 7.7. Classification accuracy for receiving devices' network connections (WiFi vs. mobile data)

Germany				UAE		
Receiver	SIG	THR	WA	Receiver	SIG	WA
DE-22	92 %	90 %	94 %	AE-22	54 %	91 %
DE-23	90 %	75 %	90 %	AE-23	61 %	89 %
DE-24	95 %	94 %	92 %	AE-24	77 %	90 %
DE-2ALL	91 %	85 %	88 %	AE-2ALL	62 %	87 %

first round of measurements, two classifications of three WiFi locations, both (b) device-specific in Germany (device *DE-23*) and (c) cross-device in the UAE (referred to as *AE-2ALL*), and (d) a receiver network classification for one of the devices (*DE-22*) in Germany. Whereas the overall classification accuracy is varying for smaller sample sizes, there are only minor improvements for more than around 100 sequences of 5 delivery confirmation timings. This observation seems to hold for all three messengers and across the different classification tasks.Thus, we can already reach considerable classification results with sample sizes of around 100 delivery confirmation timing sequences per class – for some cases, e. g., the network connection detection, even with lower sample sizes.

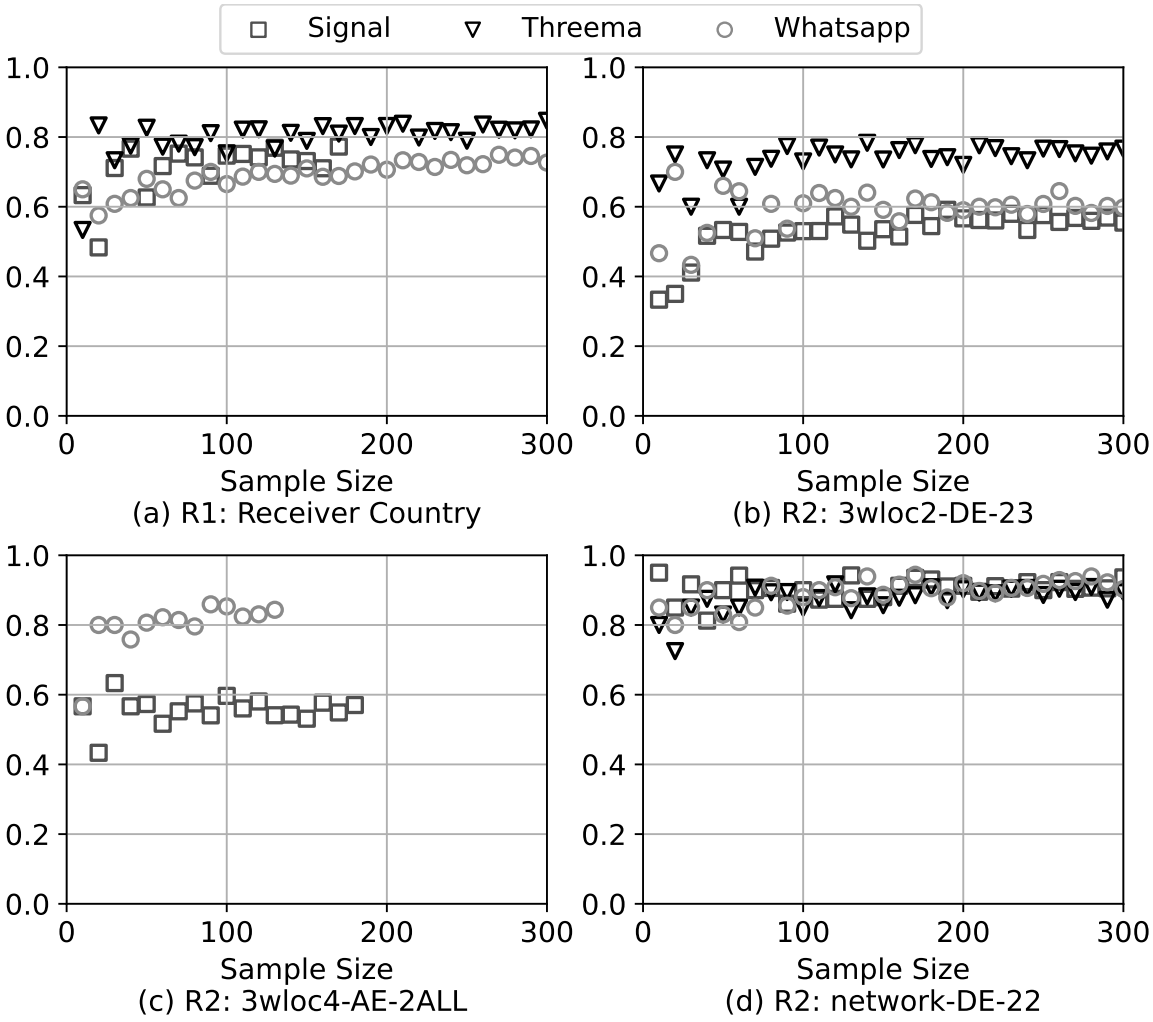

Figure 7.14. Overall accuracy of four different classification tasks, depending on the number of samples per class (x-axis).

7.5.8. Experimental Factors

While we are mostly interested in differences between receiver characteristics such as their location or network connection type, there are many dynamic features that can influence the RTTs of delivery confirmations, including network, device, and server characteristics. We now carefully discuss how such features are reflected in our measurements, and to what extent they can affect our experiments.

Network Characteristics Varying network loads, both in terms of general Internet traffic and messenger use, may affect the time required to send a message and receive the confirmations. However, such circumstances cannot be influenced by our setup. In general, network loads are mostly varying depending on the time of day, with higher loads during mornings and evenings [62, 219]. Since we continuously collect data for at least one week per receiving device and location, all relevant load levels should be covered by our measurements. When looking into our timing dataset, we do not observe large deviations or suspicious patterns depending on the time of day. Thus, the influence of network load on our dataset should be negligible.

Timings may also depend on the routes taken between sender, messenger server, and receiver, which could vary depending on the provider of the devices' Internet connections, making WiFi locations easier distinguishable when different connection providers are involved. In our measurements in Germany, only locations DE-C and DE-E were using the same connection provider but, unfortunately, our dataset does not include measurements of the same device in both locations. In the UAE, all Internet connections were provided by the same operator, with timings being fairly distinguishable (e. g., 82 % accuracy for *2wloc1-AE-22*). However, our dataset is too small, to adequately measure the effect of using the same provider at multiple locations vs. using different ones.

Device Behavior During our measurements, receiving devices were idling at each location while receiving messages. This comprises a limitation of our setup, since active interaction with the devices and parallel processes may affect the timings we measure while sending messages, with potential consequences for classification accuracy.

To overcome this issue, we conducted additional experiments over one week sending messages to one author's private smartphone while it was in everyday use and continuously recorded its network connection type (i. e., WiFi or mobile data). We then used the data to predict its network connection following the procedures described in Section 7.5.3. Classification reaches overall accuracy of 82 % for Signal, 80 % for Threema, and 74 % for WhatsApp. These numbers are fairly lower than the ones in our original and fully controlled setup (cf. Table 7.7) and shows that the threat vector still persists in a realistic usage profile, although with lower accuracy.

Server Behavior Through the experiments, the sender devices were connected to different servers when sending WhatsApp messages. We only consider WhatsApp here, since Threema only has one server location and Signal's actual infrastructure remains unclear. While the same sender connected to up to 34 different WhatsApp IP addresses (*AE-21*), 3 servers (4 for *DE-21*, respectively) make up at least 95 % of connections used when sending messages. Additionally, server selection follows similar distributions for all receiver locations. Thus, the selected server should have little unintended influence on our measurements. While our data does not contain meaningful differences in round-trip times depending on the selected server, it may be possible that strategic server selection could help the attacker (e. g., by locally changing DNS resolution) to make timings better distinguishable, i. e., further improve classification accuracy. We leave the required data collection and evaluation an open task for future work.

7.6. Countermeasures

We now shed light on possible countermeasures that can be applied to make the receiver location classification harder to better protect clients' location privacy. We consider countermeasures on the messenger's and on the user's side.

7.6.1. Randomizing Delivery Confirmation Times

Since timing measurements are a noisy source of information used for the attack, randomly delaying the delivery confirmation might be a viable solution to make timings to receivers in different locations harder to distinguish. While adding random delays must be implemented by messenger providers to come into effect, we can evaluate the impact of such a mechanism through a simulation based on the timing data we collected.

Timings can be perturbed by adding a delay sampled uniformly at random *between 0 and a specific maximum delay*. We systematically repeat the evaluation of the same four classification tasks (cf. Section 7.5.7) and increase the maximum delay in every iteration by 1 second from 0 s to 20 s. Our goal is to find a threshold value that is sufficient to make the delivery confirmation timings to receivers in different locations indistinguishable. In addition, the maximum delay should be as small as possible to keep the impact on user experience low.

Figure 7.15 shows the overall accuracy values for four classification tasks with maximum random delays between 0 s and 20 s. We selected the same classification tasks as for the classification accuracy convergence analysis, again, to cover different types of classifications (cf. Section 7.5.7). A maximum delay of 0 s corresponds to the original classification results. When we increase the maximum delay, the overall classification accuracy continuously decreases and approximates the chance of randomly guessing the location, which depends on the number of location

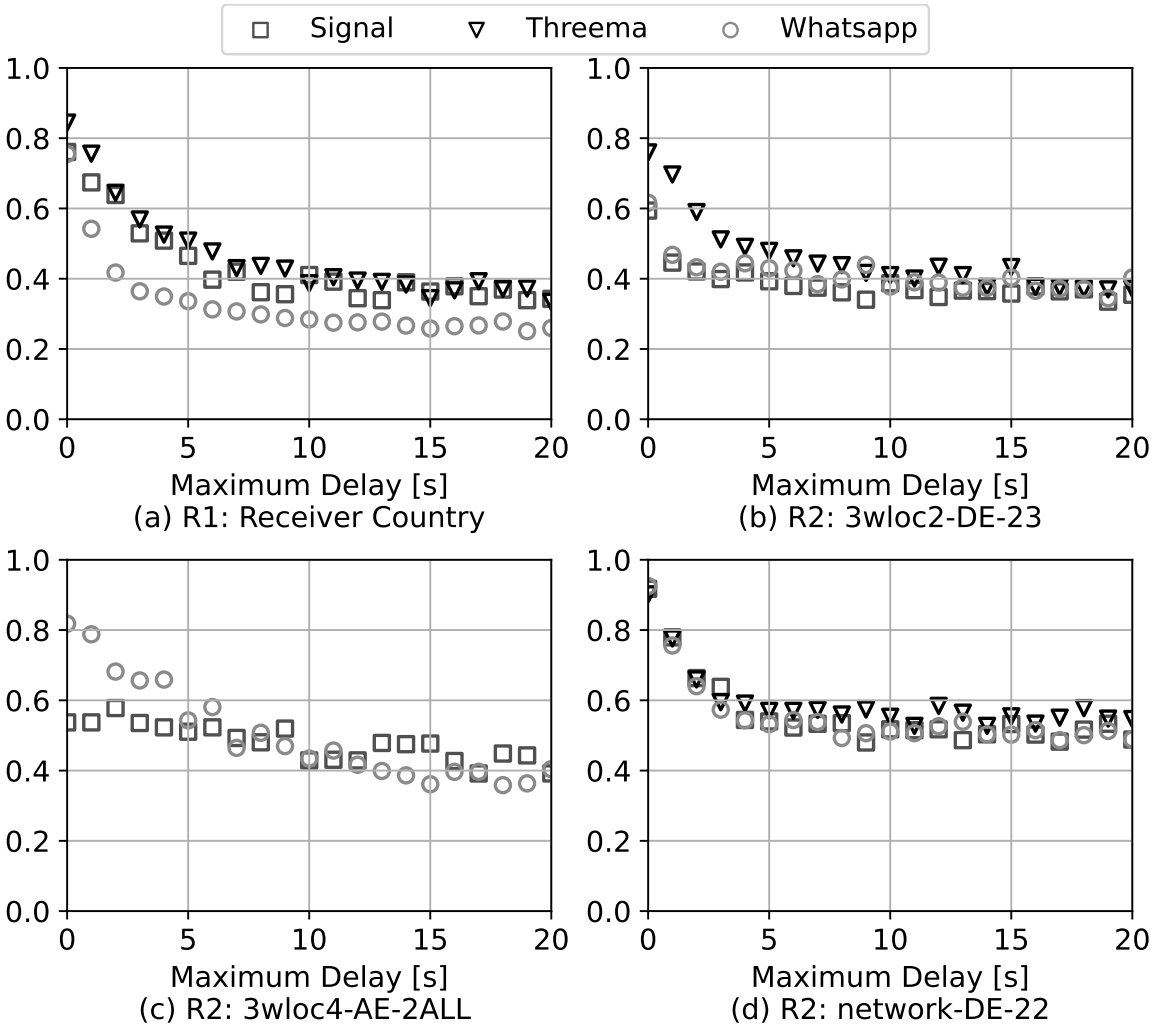

Figure 7.15. Overall accuracy of four different classification tasks with increasing random delays (x-axis) added to message delivery confirmation timings. For higher delays, the accuracy approximates the chance of randomly guessing the receiver's location.

candidates. Depending on the classification task, the random guessing accuracy is reached for a maximum delay of between 5 s and 10 s, as for example for determining the network connection of receiving device *DE-22* (cf. Figure 7.15d). Messenger servers randomly delaying delivery confirmations by up to 6 s seems to be sufficient to render the timings indistinguishable and, thus, to disable the timing side channel in message delivery confirmations. We emphasize that there is a graceful degradation of accuracy with increasing delays – introducing maximum delays

of as little as 1 or 2 seconds will already have a positive and measurable impact on users' location privacy under our attack.

If and to what extent the maximum delay can be further decreased or even flexiblized, e. g., different delays for different groups of contacts, or depending on dynamic parameters should be subject to extensive further evaluations. The best option from a user perspective would actually be the possibility to disable sending (and receiving) delivery confirmations at all – exactly as it is already offered for *read receipts* (verbatim a privacy option) in all messengers we analyzed in this paper.

7.6.2. User-side countermeasures

Users' means to reduce the effects of the timing side channel are limited, since delivery confirmations cannot be turned off in the messengers we analyzed – randomly delaying these timings can only be applied by the messenger providers. However, the use of VPN services or Tor routing all traffic through dedicated servers at distant and changing geographical locations may be a promising mitigation strategy that can be applied by users. The overhead of additional servers may perturb the delivery notifications in a similar fashion like adding random delays.

We run a small additional experiment to get a preliminary estimate of the effects of using a VPN as a countermeasure. To this end, we send messages to one receiver phone (*DE-23*) in one location (*DE-B*) both on WiFi and cellular Internet connections – in both cases connected to a US-based VPN server provided by a commercial VPN provider. Whereas without VPN, the network connection of this device can be distinguished with up to 90 % accuracy (cf. Table 7.7, classifications perform worse when using a VPN. For Threema (51 %) and WhatsApp (62 %), performance is hardly better than random guessing (50 %). However, for Signal, we reach a surprisingly high overall accuracy of 77 %. When repeating the same small experiment with using Tor instead of a VPN,

WiFi and cellular connections can be distinguished better (Signal: 72 %, Threema: 58 %, WhatsApp: 82 %).

Without investigating these issues more systematically, we can only speculate about the reasons. One explanation could be that Signal's servers are US-based and, therefore, the routing overhead introduced by using the VPN server is too small to adequately perturb timings. For the case of Tor, the set of circuits selected in either sample may have biased the comparably small sets of timings we measured. However, since conclusive statements require more systematic and extensive measurements to allow a thorough evaluation, we leave this issue an open task for future work.

Since users' means to perturb timings and, thus, to disable the side channel seem ineffective in practice, another option could be to totally block delivery confirmations, e. g., by filtering the related packets based on their size out of their local network traffic by means of a firewall. While this might be a viable solution for technically adept users or in specifically security-sensitive use cases, it does, however, not apply to the vast majority of the 2 billion WhatsApp users.

7.7. Conclusion

We presented a novel timing side-channel in popular instant messengers, allowing to distinguish different receivers and their locations by sending them instant messages. We have demonstrated how measuring the time between sending a message and receiving the notification that the message has been delivered enables clients to spy on each other, e. g., to determine whether or not they are at their usual location. While making use of this side channel is mostly limited to people who are in each others' contact lists and have already started a conversation before, it yet comprises an unexpected and privacy-infringing act with low technical requirements that is equally hard to detect and to mitigate for a potential victim.

8
Conclusion

Contents

8.1. Summary and Key Results

In this thesis, we provided a broad analysis of different types of information that is exposed within applications and on the Internet when users interact with digital communication applications. As we have seen, users do not only share data intentionally but also the use of specific applications can reveal information about them to others. For information that is intentionally made available to others, we mainly focused on exposure reduction features, eventually resulting in data lifetime ending. In the context of information that users unintentionally reveal, we demonstrated two types of applications. Tor represents technology that is explicitly used for privacy purposes such as concealing one's identity and messenger apps represent ever-present tools that are widely adopted and used by billions of users for everyday communication purposes.

8.1.1. Managing Self-Published Online Data

In our systematic review of longitudinal online data management in Chapter 2, we categorized a broad range of technical approaches for managing online data longitudinally and studies analyzing how users interact with such features in existing applications and environments. By contrasting technical and user side, we identified incorrect, incomplete, and missing realizations of users' desires in academic proposals for technical solutions. Based on such conflicts, we then derived a set of technical key challenges evolving around the need for flexibilization of data lifetime ending and its conditions, and to better incorporate user perception of security and trust, and their mental models associated with it. The challenges we identified can serve as recommendations for the development of new mechanisms for users to manage their information exposure online.

In a mobile messaging context, we then explored users' perception of and preferences for message deletion options in messengers in Chapter 3. We particularly focused on whether users were able to assess if messages

were only deleted on their own device or also from devices of recipients. We initiated our study following the roll-out of a new feature in WhatsApp, in which users could explicitly select between these two options. Our initial assumption that the effects of message deletion were ambiguous without a clear choice was confirmed by our preparatory analysis of deletion functions in 17 messengers. In our study, we found that users could better determine where a message was deleted when the effects of deletion were explained, as it was implemented in the newly introduced feature in WhatsApp. Our results show that subtleties such as the integration of a simple dialogue have the potential to facilitate user understanding of app functionality and, thus, improve user experience when handling the data they made available to others.

In order to widen the views and also incorporate other non-technical perspectives to allow for more flexibility in the specification of exposure control mechanisms, we proposed a solution combining technical and legal aspects in Chapter 4. Our approach enables users and online platforms to agree on conditions for reducing exposure of online data up to entirely removing it from public access. In order to incentivize the providers of online platforms to comply with the agreement, we used a penalty mechanism that can be triggered by the users and is verified by a neutral authority. Our prototype implementation involving smart contracts based on the Ethereum cryptocurrency system shows that our concept is technically feasible. It also demonstrates how completely new approaches to controlling one's online exposure, designed from scratch, could work.

8.1.2. Usage-Driven Information Revelation

In Chapter 6, we analyzed the feasibility of traffic analysis attacks on Tor, i. e., revealing users' identities and the services they use, under real-world conditions. Compared to theoretically perfect attack performance, real-world adversaries do only have access to a limited set of Tor relays

and can only deanonymize Tor users when they use relays under adversarial control. In this context, we developed three novel stepping-stone attacks that have the potential to reduce the efforts for adversaries as well as improving their chances to uncover Tor users' identities. We have shown how adversaries can use a timing side channel in the circuit establishment handshake to predict the exit relay of the connection to determine in advance whether traffic analysis can be successful. Additionally, adversaries can actively interfere with the circuit establishment to improve their chances for successful user deanonymization. Since all attacks exploit core defensive mechanisms of Tor's circuit establishment, there is no simple way to mitigate them. One possible countermeasure includes obfuscating timings, i. e., adding artificial delays within the circuit establishment procedure, which reduces the attack performance but comes at the cost of usability.

In Chapter 7, we have shown how the whereabouts of individuals can be leaked by exploiting a surprising timing side-channel in widely adopted everyday communication applications. Messenger users can spy on their contacts by simply sending them instant messages and observing the time it takes for message delivery to be confirmed. Timing distributions differ between locations, most likely due to characteristics of the respective Internet connection. Thus, after an initial training phase, an adversary can send a target user a sequence of five messages and determine their whereabouts with up to 95 % accuracy in the scenarios we evaluated. While accuracy varies between scenarios and the three messengers we tested, our results imply that the side-channel exists independent of the underlying messenger infrastructure.

8.2. Directions for Future Research

In the following, we point towards a set of potential topics that we identified while carrying out the work presented in this thesis and that we consider interesting to be addressed by future research.

8.2.1. Filling Gaps Between Technical and User Research

The basis to determine subsequent research tasks is provided by our systematization in Chapter 2. The challenges we identified comprise gaps between technical and user-centered research and each directly points towards open issues that can only be appropriately resolved by taking both sides into account.

For example, incomplete realizations of expiration or exposure reduction mechanisms provide strong indications that users' intentions have not been appropriately addressed. Whereas it is technically sound to develop mechanisms that let data entirely disappear, such mechanisms neglect users more fine-grained preferences, e. g., to make data unavailable for a general audience on the one hand, while at the same time keeping it available for their core peer group.

Therefore, studying users attitudes towards protecting their online data is inevitable in the process of developing new technology. Only when users' intentions are entirely clear, research can equip them with useful tools they need for controlling their information exposure.

8.2.2. Focus on Improving Existing Applications

Developing sound and provably secure concepts and protocols for features such as data deletion represents important foundational work. However, the path to bringing such new mechanisms into effect is not taken by implementing a ground-breaking new tool providing the respective technology and waiting for users to adopt it. Instead, providing better systems most likely entails extending existing applications with additional features that satisfy users needs in interacting with these applications and – ideally – come into effect by default. A related textbook example for this method was showcased by WhatsApp, adapting the Signal protocol for end-to-end encrypted message exchange and turning it on by

default. This procedure effectively enabled end-to-end encrypted communication for more than one billion users world-wide within a moment, most likely without users even noticing.

Following this example, research should explore how newly developed mechanisms for information exposure control are compatible with and can be integrated into popular and widely-adopted applications. This way, many users can immediately profit from latest progress, ideally without being forced to change their behavior, and likewise, new developments can easier find adoption among relevant audiences.

8.2.3. Practicality of Data Revocation Contracts

Our proposal to use agreements based on smart contracts as a means for online data revocation widened the perspectives onto the topic of information exposure control by incorporating legal aspects. While we have presented the fundamental concepts for interaction between users and providers at different stages in the data lifecycle, and demonstrated its technical feasibility with a prototype implementation, the user perspective remains yet unclear.

Since we have sketched how contracts allow for more flexible exposure control mechanisms, we assume that they have the potential to point into the right direction and can better fulfill users' desires in handling their online data. However, for such a mechanism to be eventually deployed in a practical environment, additional research studying users' willingness to adopt it is necessary.

Moreover, the use of currently available cryptocurrency systems blockchains entails additional questions regarding the excessive energy consumption of computationally expensive proof-of-work blockchains. Thus, introducing data revocation smart contracts requires a lot of additional research on the sustainability of the underlying technology and its societal impact, or otherwise comprises a severe burden for currently unresolved environmental challenges.

8.2.4. Trade-offs Between Privacy and Usability

Our technical analyses in the second part of this thesis have showcased two examples for information exposure in digital communication environments unintended by users. For the case of Tor traffic analysis, we have demonstrated how randomized timing delays can help to reduce the attacker success in uncovering users' identities, eventually rendering the attack useless. However, artificially delaying Tor's service comes at the cost of degrading user experience which has also been acknowledged before [67].

Future research could study to what degree users' are willing to tolerate usability cutbacks in exchange for additional privacy or better control of their information exposure. Specifically for the case of privacy-focused applications such as Tor, we discussed ideas evolving around letting users choose between usability and privacy but leave it an open task for future work to explore how such mechanisms could be realized and to study if and how users would be willing to interact with it.

While we have provided evidence that timing delays can indeed mitigate the unintended identity leaks in Tor, such countermeasures might also be effective for the location exposure in messengers which needs to be investigated in practice. Yet again, this entails the need to study potential usability issues, since setting up countermeasures with effects on user experience should not take place without capturing users attitudes towards it before.

Similar to the challenges regarding active data sharing in the first part, we again emphasize the need for joint research from multiple perspectives to address and resolve practical issues as a whole.

8.2.5. Practicality of Traffic Analysis in Messengers

For the location revelation in messenger apps, we have focused on demonstrating its feasibility from a purely technical perspective. Whereas we have shown high accuracy for the location prediction in our experimental

setup, it must be further explored how our results translate into situations in the real world.

Particularly the phase in which the adversary learns the timing patterns of different locations of a contact is presumably trickier to realize than in our experimental setting. Whereas it is not uncommon for contacts to regularly exchange messages, repeatedly sending the same message sequence at constant time intervals will most likely be considered suspicious by a potential target, before any meaningful data could have been collected.

Therefore, studying users might be helpful to determine to what degree sending messages must be throttled for the attack to remain undetected. Additionally, future research could examine if the respective timings could also be collected alongside regular conversations. Actual user behavior is, however, harder to simulate in the lab but reflecting it is a necessary step to fully assess the actual threat under real-world conditions.

8.3. Closing Remarks

In a broader sense, the work presented in this thesis has demonstrated that isolated views on practical topics involving users and applications they are actually using only from a technical security perspective is not enough to build better systems or to improve existing ones. We again emphasize that research studying digital applications that users actually use should always incorporate perspectives from multiple disciplines in order to allow for designing systems and mechanisms that are a benefit for users. This is inevitable to allow for providing technology users actually need and that can help them to better control their own information exposure in digital communication environments.

List of Figures

List of Tables

A

User Perception of Message Deletion

This appendix for Chapter 3 includes the survey instrument used for the study and the study results

A.1. Survey Instrument

A.1.1. Instructions and Privacy Statement

This study by Ruhr University Bochum's Mobile Security Group investigates how users use and perceive the deleting functionality in instant messengers on mobile devices. We just asked you to write and delete a message in an instant messaging app. This survey will ask you some questions about how you use the "delete message" feature in mobile instant messengers and how you expect this feature to work. If you have any questions about the survey, feel free to ask any time! Privacy Policy: All data we collect in the course of this study is treated confidentially. We store all the answers you have entered for further evaluation and analysis. We also measure the total time it takes you to complete the survey and perform the experimental tasks. All data we have collected is stored anonymously such that it is not possible to connect the data to your person at any point in time. Please note that your choice to participate in this study is completely voluntary. You are free to withdraw from the study at any time, and we will discard all of your data and not analyze or store it. If you agree with this procedure, click the *Next* button to begin.

A.1.2. Questions

Q1) Do you frequently use instant messaging (e.g., WhatsApp or Snapchat) on a mobile device (e. g., smart phone or tablet computer)? (*frequently means several times a month*)

- Yes
- No

Q2) Which mobile operating systems do you use? (Multiple answers possible)

- Android
- iOS
- Windows Phone
- Other

Q3) Which instant messaging services do you use? (Multiple answers possible)

- Facebook Messenger
- Google Hangouts
- GroupMe
- Line
- Apple Messages
- QQ Mobile
- Signal
- Skype
- Snapchat
- Telegram
- Threema
- Viber
- WeChat
- WhatsApp
- Other

Q4) How often do you delete instant messages?

- Several times a day
- About once a day
- A few times a week

- A few times a month
- A few times a year
- Almost never
- I don't know

Q5) What are your reasons for deleting messages?

Free text

Q6) We just asked you to send a message and then to delete it. What do you think—where has the message been deleted?

- From the sender's device
- From the recipient's device
- Other

Q7) Which of the following do you prefer when you delete a message?

- The message is deleted from my device only.
- The message is deleted from recipient's device only.
- The message is deleted from both devices.
- For each message, I can choose where to delete the message from.

Q8) Do you want to be notified if the recipient has already read the message?

- Yes
- No

Q9) Do you think that the recipient should be told that the message has been deleted (e.g., through a "message deleted" hint)?

- Yes
- No

Q10) How old are you?

- <20
- 20–34
- 35–49
- ≥50
- No answer

Q11) With which gender do you identify?

- Female
- Male
- Other
- No answer

Q12) Please estimate your level of experience with mobile devices. (1—Beginner; 5—Expert)

- 1 (beginner)
- 2
- 3
- 4
- 5 (expert)
- No answer

Q13) Does this result match your expectations?

- Yes
- No

Q14) Why does this result match your expectations? Why not?

Free text

Q15) Do you think the delete function should be limited (e.g., only messages of the last hour, only the latest message, only unread messages could be deleted)?

- Yes
- No

Q16) How should the delete function be limited? Please specify.

Free text

A.2. Results

	WhatsApp	Skype	Facebook Messenger	All Conditions
Q1) Do you frequently use instant messaging?				
Yes	38	38	44	120
No	1	3	1	5
Q2) Do you frequently use instant messaging?				
Android	26	17	33	76
iOS	12	22	15	49
Windows Phone	6	1	1	8
Other	1	1	1	3
Q3) Which instant messaging services do you use?				
Facebook Messenger	20	22	21	63
Google Hangouts	1	3	2	6
GroupMe	0	0	0	0
Line	0	1	0	1
Apple Messages	9	9	5	23
QQ Mobile	1	1	1	3
Signal	4	6	4	14
Skype	15	12	9	36
Snapchat	10	12	11	33
Telegram	11	19	11	41
Threema	3	6	4	13
Viber	4	3	1	8
WeChat	2	3	2	7
WhatsApp	34	39	41	114
Other	3	4	6	13
Q4) How often do you delete instant messages?				
Several times a day	2	3	4	9
About once a day	1	1	2	4
A few times a week	3	2	4	9
A few times a month	4	6	3	13

Continued on next page

Table A.1 – continued from previous page

	WhatsApp	**Skype**	**Facebook Messenger**	**All Conditions**
A few times a year	4	4	6	14
Almost never	23	24	24	71
I don't know	2	1	2	5
Q6) [...] Where has the message been deleted?				
Sender	32	35	45	112
Recipient	29	12	7	48
Other	4	4	1	9
Q7) Which [...] do you prefer when you delete a message?				
Sender only	3	3	6	12
Recipient only	2	3	3	8
Both devices	16	19	19	54
Select	18	16	17	51
Q8) Do you want to be notified if the recipient has already read the message?				
Yes	35	31	31	97
No	4	10	14	28
Q9) Do you think that the recipient should be told [...]?				
Yes	13	17	16	46
No	26	24	29	79
Q10) How old are you?				
<20	6	10	9	25
20–34	32	26	30	88
35–49	1	3	4	8
≥ 50	0	2	1	3
No answer	0	0	1	1
Q11) With which gender do you identify?				
Female	17	12	11	40
Male	21	28	31	80
Other	0	1	2	3
No answer	1	0	1	2
Q12) [...] level of experience with mobile devices				

Continued on next page

Table A.1 – continued from previous page

	WhatsApp	**Skype**	**Facebook Messenger**	**All Conditions**
1 (beginner)	0	1	0	1
2	3	3	5	11
3	7	11	12	30
4	16	15	19	50
5 (expert)	11	10	7	28
No answer	2	1	2	5
Q13) Does this result match your expectations?				
Yes	31	20	32	83
No	8	21	13	42
Q15) Do you think the delete function should be limited [...]?				
Yes	10	16	13	39
No	29	25	32	86

B

Operational Requirements for Tor Traffic Analysis

This appendix for Chapter 6 includes detailed exit prediction performance results (cf. Figure 6.6) for the top 10 countries in exit bandwidth as nation-state adversaries.

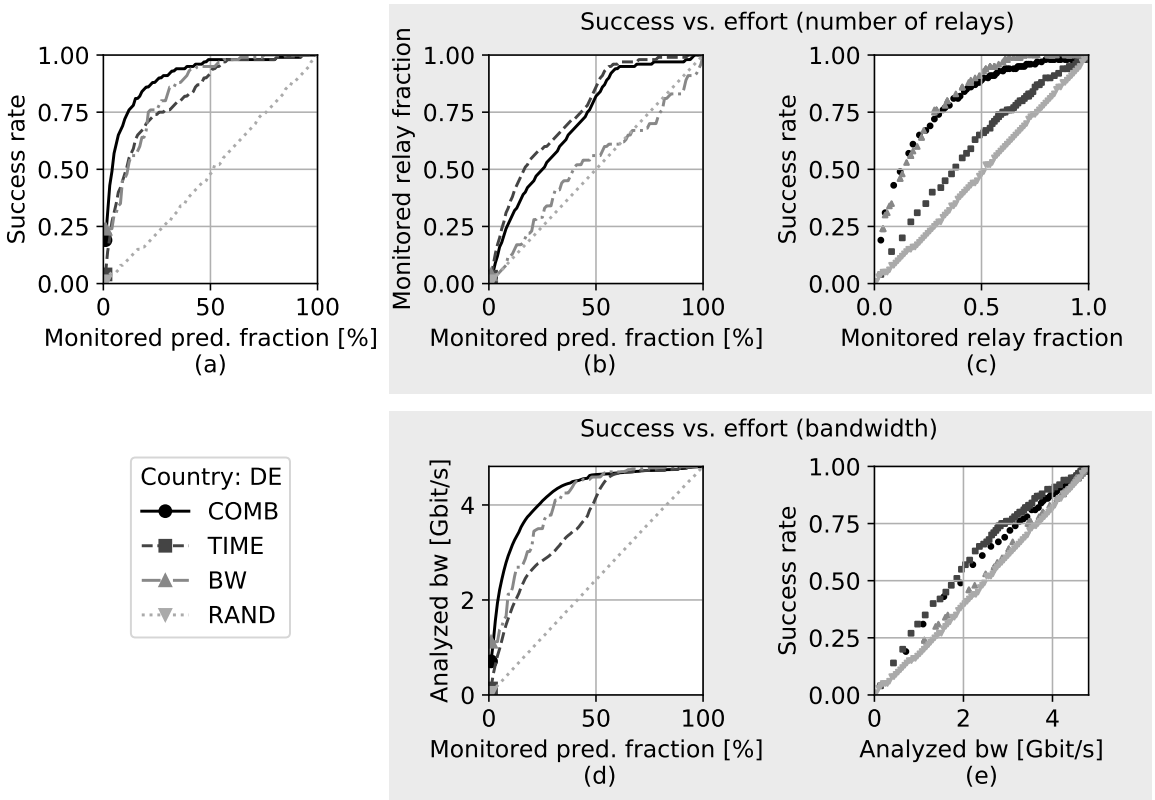

Figure B.1. Exit Prediction Performance for the DE adversary.

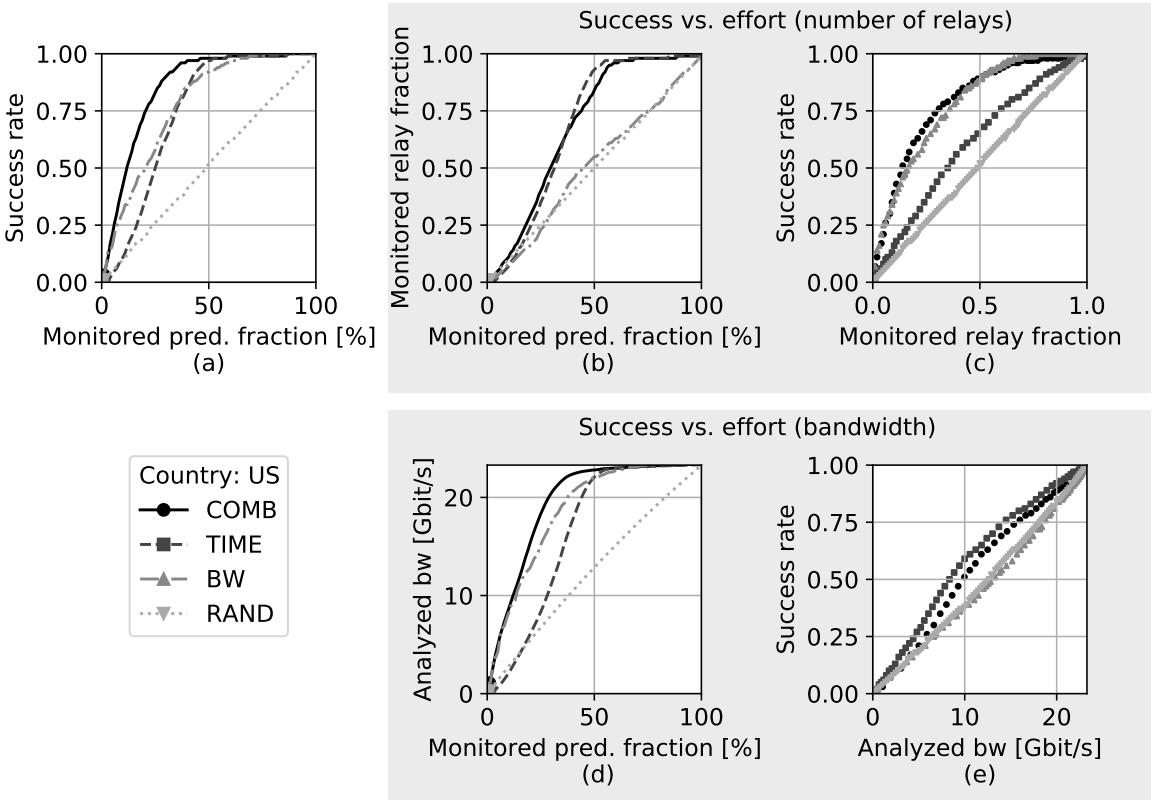

Figure B.2. Exit Prediction Performance for the US adversary.

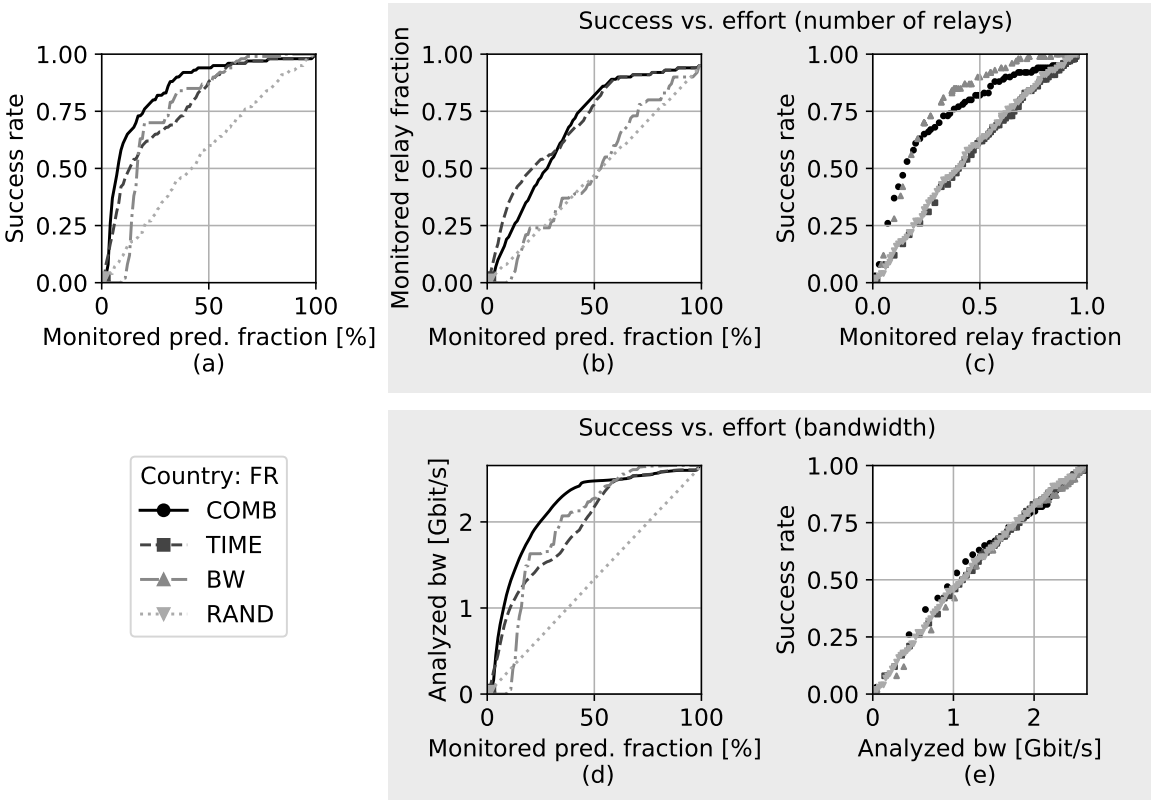

Figure B.3. Exit Prediction Performance for the FR adversary.

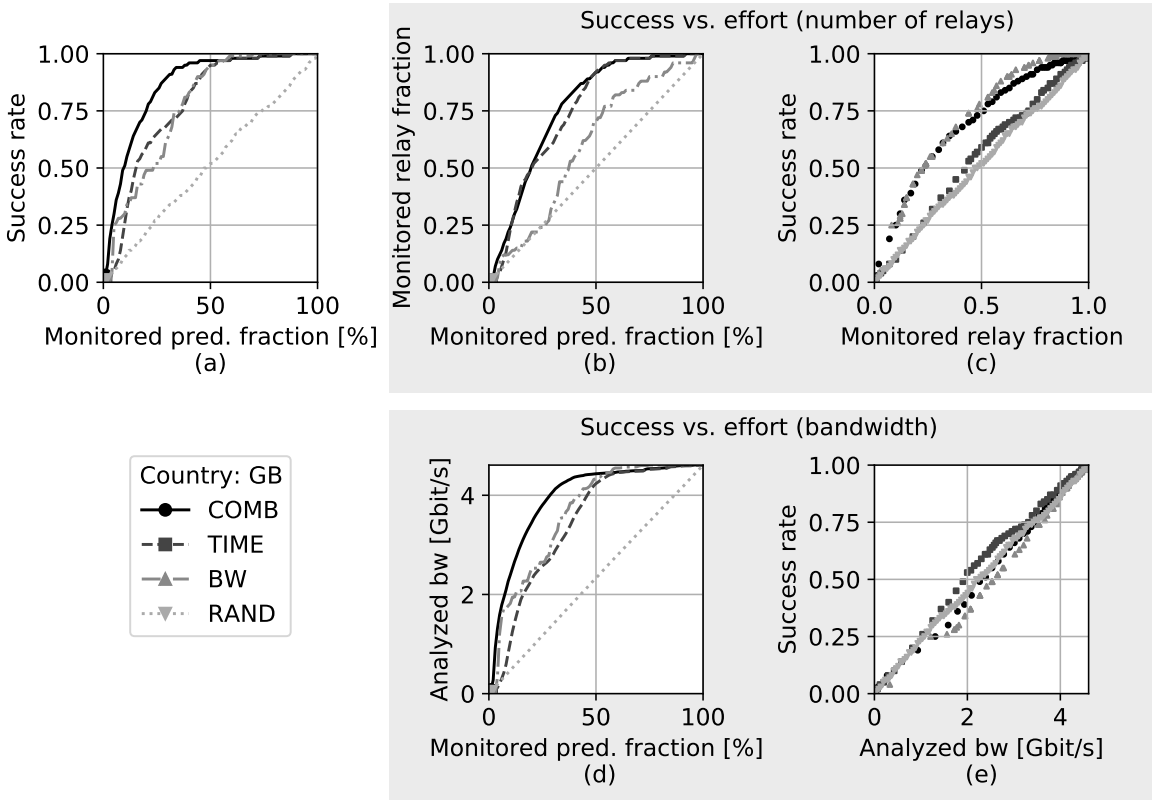

Figure B.4. Exit Prediction Performance for the GB adversary.

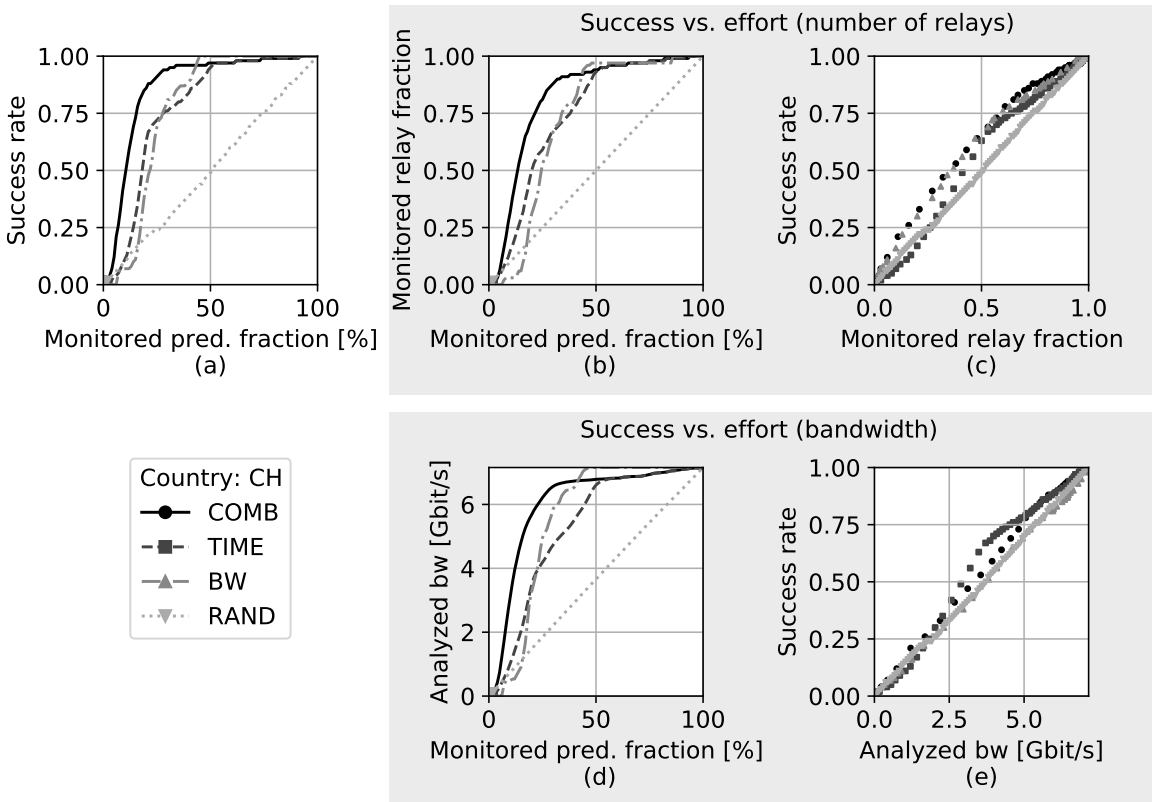

Figure B.5. Exit Prediction Performance for the CH adversary.

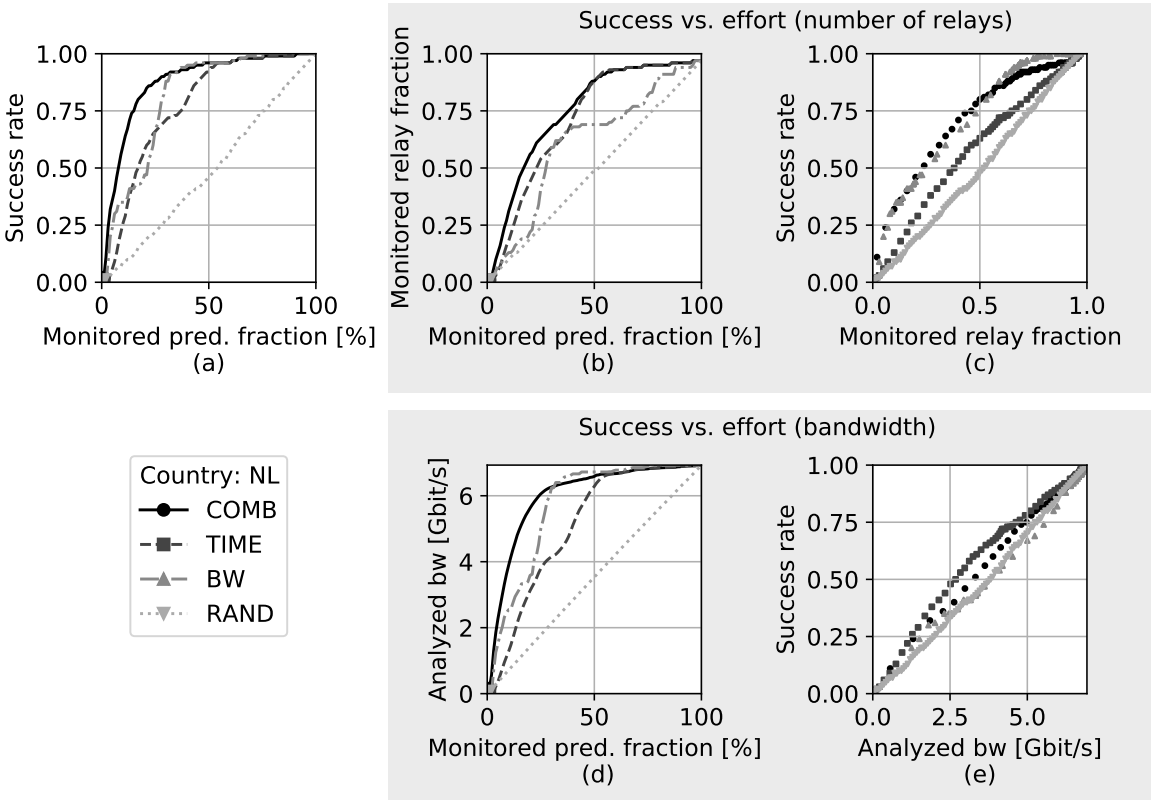

Figure B.6. Exit Prediction Performance for the NL adversary.

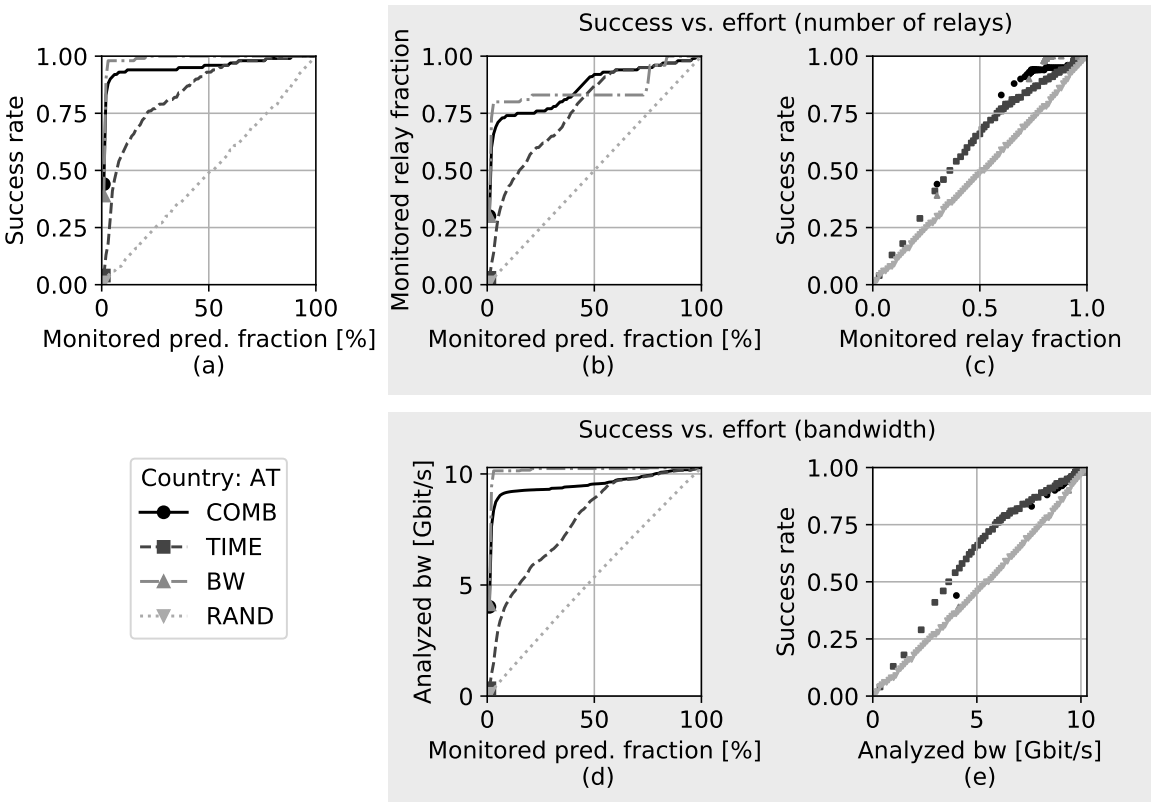

Figure B.7. Exit Prediction Performance for the AT adversary.

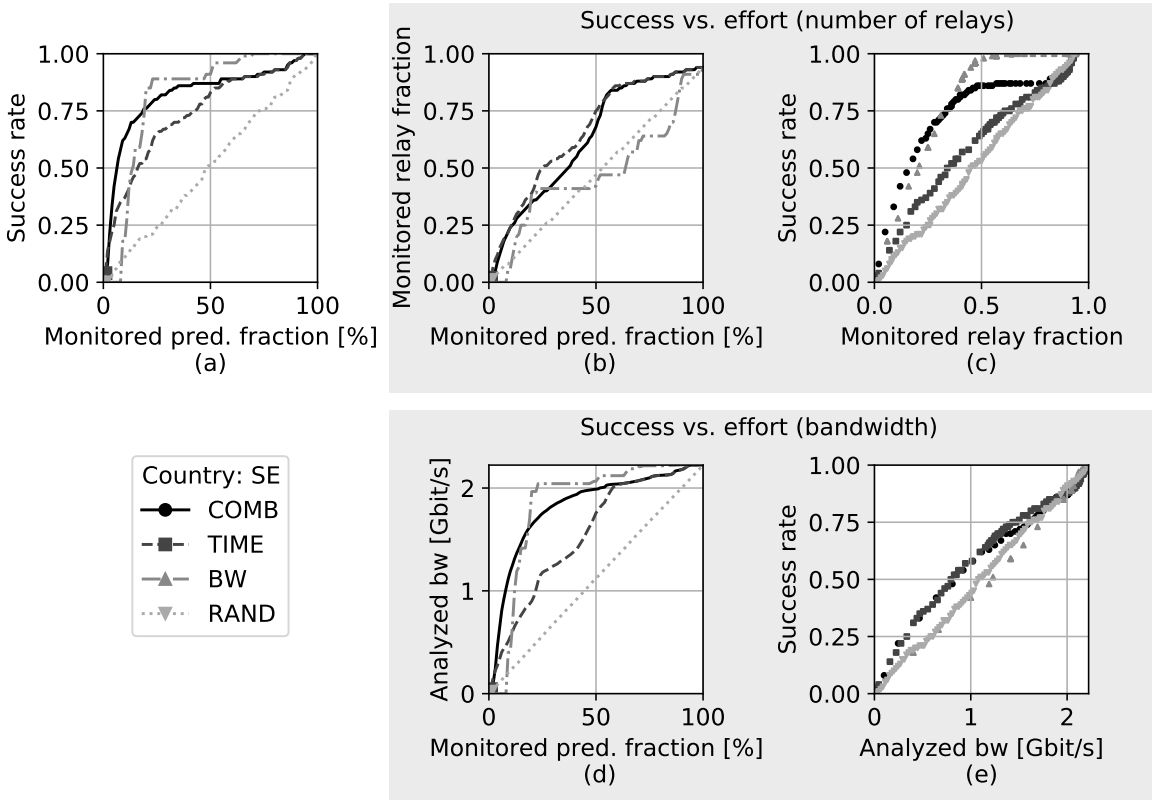

Figure B.8. Exit Prediction Performance for the SE adversary.

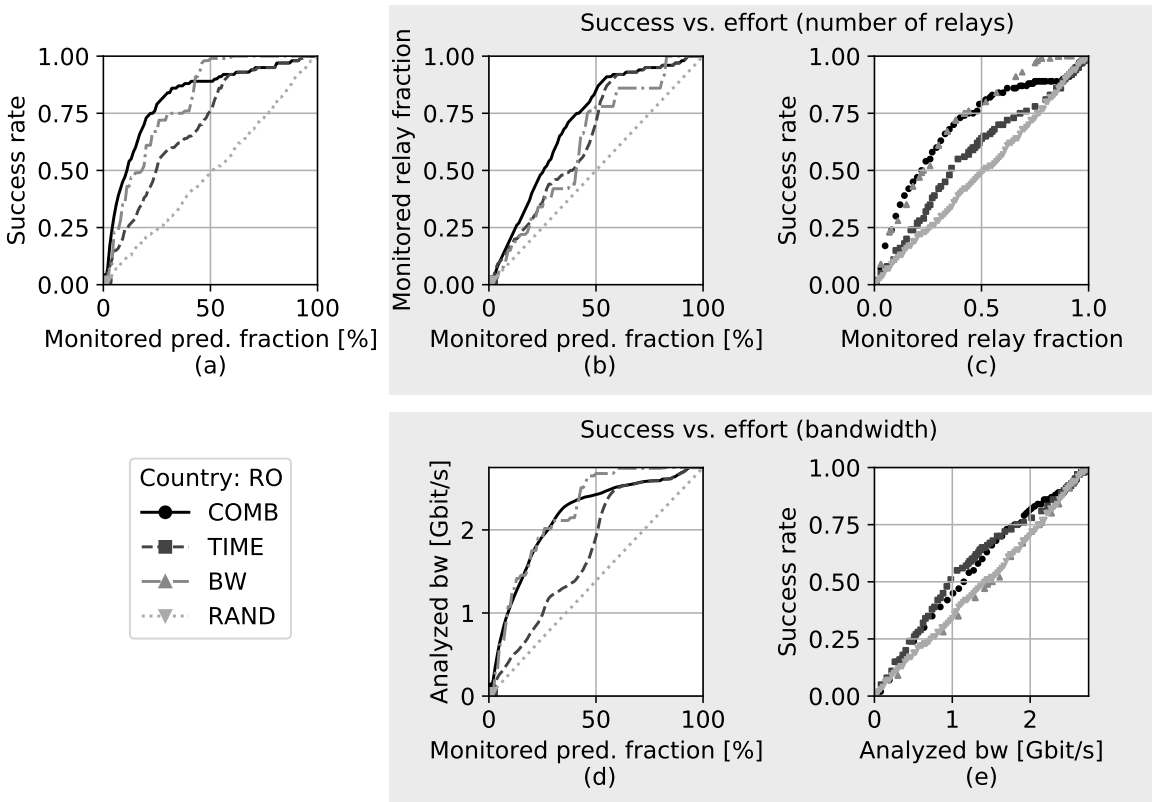

Figure B.9. Exit Prediction Performance for the RO adversary.

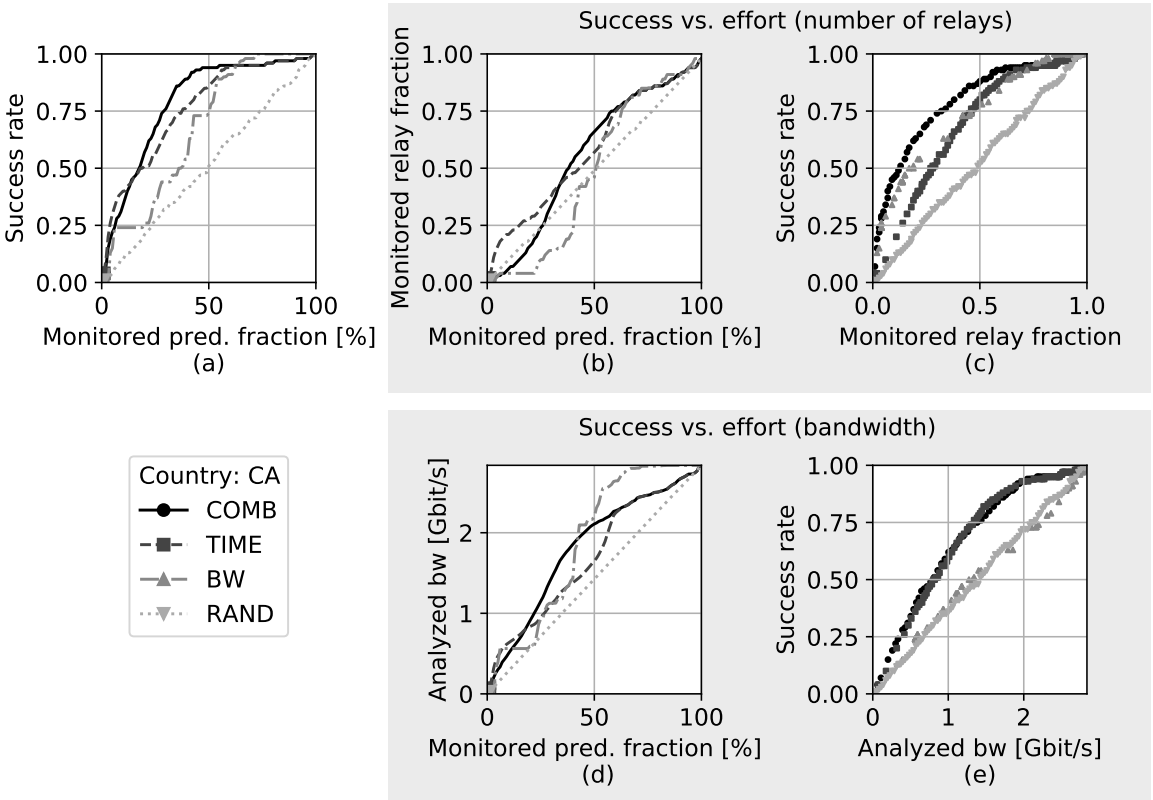

Figure B.10. Exit Prediction Performance for the CA adversary.

C
Location Revelation in Instant Messengers

This appendix for Chapter 7 includes detailed results for all instances of all classification tasks. The tables report precision values for each class and overall classification accuracy (*All*). Five values per messenger represent different notification sequence lengths.

C.1. Round 1

Table C.1. Detailed classification results for the first round of measurements.

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Two countries measured with sender DE-11																
DE11-2countries1	All	.81	0.83	.85	.85	.91	.73	.81	.84	.86	.92	.77	.85	.89	.9	.91
DE11-2countries1	DE	.77	.81	.88	.86	.89	.83	.82	.83	.89	.91	.84	.83	.88	.89	.9
DE11-2countries1	NL	.84	.86	.82	.85	.92	.63	.79	.84	.84	.93	.7	.87	.91	.91	.93
DE11-2countries2	All											.82	.85	.85	.86	.95
DE11-2countries2	AE											.88	.9	.87	.91	.96
DE11-2countries2	DE											.76	.8	.84	.8	.94
DE11-2countries3	All	.76	.85	.87	.88	.89	.87	.9	.92	.95	.96	.77	.83	.84	.86	.9
DE11-2countries3	DE	.75	.81	.86	.9	.88	.89	.91	.93	.94	.96	.76	.85	.88	.9	.91
DE11-2countries3	GR	.76	.88	.87	.87	.9	.84	.89	.91	.95	.96	.77	.81	.81	.83	.89
DE11-2countries4	All											.66	.78	.82	.85	.89
DE11-2countries4	AE											.57	.75	.8	.85	.88
DE11-2countries4	NL											.75	.81	.84	.85	.9

Continued on next page

Table C.1 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
DE11-2countries5	All	.64	.65	.7	.71	.76	.68	.75	.77	.82	.85	.57	.65	.7	.73	.78
DE11-2countries5	GR	.83	.74	.78	.73	.71	.76	.67	.76	.83	.89	.83	.63	.68	.74	.79
DE11-2countries5	NL	.45	.57	.61	.7	.82	.6	.83	.78	.8	.8	.3	.67	.72	.73	.77
DE11-2countries6	All											.66	.79	.83	.86	.92
DE11-2countries6	AE											.56	.77	.81	.86	.9
DE11-2countries6	GR											.77	.81	.86	.87	.93
Three countries measured with sender DE-11																
DE11-3countries1	All											.61	.74	.77	.78	.87
DE11-3countries1	AE											.52	.74	.73	.84	.88
DE11-3countries1	DE											.75	.73	.75	.72	.88
DE11-3countries1	NL											.56	.76	.83	.78	.85
DE11-3countries2	All	.6	.65	.69	.7	.79	.6	.7	.73	.79	.86	.54	.63	.69	.73	.77
DE11-3countries2	DE	.76	.75	.76	.87	.9	.81	.81	.8	.88	.94	.77	.78	.85	.86	.83
DE11-3countries2	GR	.6	.65	.71	.7	.77	.77	.77	.78	.79	.84	.71	.52	.51	.65	.73
DE11-3countries2	NL	.44	.54	.59	.54	.7	.23	.52	.6	.69	.8	.14	.6	.69	.69	.75
DE11-3countries3	All											.62	.72	.76	.77	.87
DE11-3countries3	AE											.59	.71	.77	.83	.91
DE11-3countries3	DE											.75	.73	.77	.72	.88
DE11-3countries3	GR											.51	.71	.73	.76	.81
DE11-3countries4	All											.45	.61	.68	.73	.78
DE11-3countries4	AE											.49	.67	.76	.79	.84
DE11-3countries4	GR											.57	.58	.69	.69	.77
DE11-3countries4	NL											.28	.58	.59	.7	.73
Four countries measured with sender DE-12																
DE11-4countries	All											.48	.6	.64	.67	.77
DE11-4countries	AE											.5	.64	.72	.74	.84
DE11-4countries	DE											.73	.71	.7	.72	.81
DE11-4countries	GR											.55	.53	.57	.58	.79
DE11-4countries	NL											.13	.51	.55	.66	.63
Within-country classification measured with sender DE-11																
DE11-within-ae	All											.69	.79	.81	.84	.91
DE11-within-ae	AE											.74	.82	.85	.85	.9
DE11-within-ae	NOT-AE											.63	.76	.77	.83	.91

Continued on next page

Table C.1 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
DE11-within-de	All	.79	.82	.86	.88	.91	.8	.85	.87	.9	.92	.78	.82	.83	.85	.91
DE11-within-de	DE	.73	.82	.85	.88	.92	.88	.86	.88	.91	.91	.77	.78	.81	.84	.9
DE11-within-de	NOT-DE	.85	.83	.87	.87	.91	.72	.85	.85	.88	.94	.79	.86	.84	.85	.92
DE11-within-gr	All	.69	.77	.76	.81	.85	.8	.83	.82	.86	.88	.65	.74	.75	.78	.83
DE11-within-gr	GR	.72	.88	.89	.85	.91	.87	.86	.85	.87	.89	.66	.73	.77	.75	.83
DE11-within-gr	NOT-GR	.67	.66	.64	.77	.79	.72	.79	.8	.86	.87	.65	.74	.74	.81	.83
DE11-within-nl	All	.78	.79	.83	.83	.85	.61	.72	.75	.79	.85	.68	.76	.79	.8	.82
DE11-within-nl	NL	.85	.86	.83	.85	.85	.65	.77	.76	.83	.88	.65	.84	.88	.85	.83
DE11-within-nl	NOT-NL	.71	.71	.83	.82	.84	.57	.66	.74	.74	.83	.7	.68	.69	.74	.81
Two countries measured with sender DE-12																
DE12-2countries1	All	.75	.84	.87	.84	.88	.67	.77	.86	.86	.86	.72	.8	.86	.87	.81
DE12-2countries1	DE	.83	.8	.84	.82	.9	.73	.81	.84	.87	.87	.75	.79	.88	.87	.82
DE12-2countries1	NL	.66	.88	.9	.86	.86	.62	.74	.87	.86	.86	.69	.8	.84	.87	.8
DE12-2countries2	All											.86	.92	.94	.96	.98
DE12-2countries2	AE											.95	.97	.98	.97	.97
DE12-2countries2	DE											.77	.88	.9	.95	.99
DE12-2countries3	All	.59	.62	.62	.6	.67	.63	.77	.79	.86	.78	.76	.81	.84	.82	.85
DE12-2countries3	DE	.73	.77	.68	.6	.67	.75	.76	.76	.86	.78	.84	.83	.87	.88	.87
DE12-2countries3	GR	.44	.46	.57	.61	.67	.51	.78	.83	.86	.77	.68	.78	.81	.76	.84
DE12-2countries4	All											.77	.87	.91	.92	.94
DE12-2countries4	AE											.85	.88	.95	.92	.94
DE12-2countries4	NL											.7	.85	.87	.93	.94
DE12-2countries5	All	.61	.75	.74	.82	.78	.55	.7	.78	.8	.82	.62	.58	.59	.57	.73
DE12-2countries5	GR	.49	.62	.72	.84	.76	.31	.62	.68	.77	.87	.42	.48	.59	.54	.77
DE12-2countries5	NL	.74	.88	.75	.81	.8	.8	.78	.88	.84	.78	.82	.68	.6	.6	.68
DE12-2countries6	All											.82	.89	.95	.94	.96
DE12-2countries6	AE											.87	.89	.96	.93	.96
DE12-2countries6	GR											.78	.9	.95	.94	.96
Three countries measured with sender DE-12																
DE12-3countries1	All											.66	.76	.78	.84	.88
DE12-3countries1	AE											.77	.89	.87	.91	.94
DE12-3countries1	DE											.71	.74	.75	.86	.86
DE12-3countries1	NL											.51	.65	.73	.74	.83
DE12-3countries2	All	.46	.5	.49	.54	.62	.45	.57	.69	.72	.69	.57	.59	.62	.61	.71

Continued on next page

Table C.1 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
DE12-3countries2	DE	.76	.64	.44	.69	.73	.72	.68	.73	.72	.7	.71	.76	.77	.79	.77
DE12-3countries2	GR	.01	.17	.4	.27	.53	.13	.37	.53	.67	.68	.43	.48	.52	.57	.7
DE12-3countries2	NL	.6	.68	.63	.66	.61	.51	.67	.8	.77	.68	.57	.52	.57	.47	.64
DE12-3countries3	All											.7	.77	.85	.85	.89
DE12-3countries3	AE											.84	.84	.94	.94	.95
DE12-3countries3	DE											.69	.71	.82	.84	.86
DE12-3countries3	GR											.57	.76	.79	.79	.86
DE12-3countries4	All											.55	.68	.67	.71	.72
DE12-3countries4	AE											.79	.87	.9	.91	.94
DE12-3countries4	GR											.54	.64	.52	.54	.64
DE12-3countries4	NL											.31	.52	.59	.69	.59
Four countries measured with sender DE-12																
DE12-4countries	All											.57	.59	.66	.67	.72
DE12-4countries	AE											.83	.83	.87	.93	.94
DE12-4countries	DE											.72	.64	.73	.84	.77
DE12-4countries	GR											.45	.55	.53	.52	.64
DE12-4countries	NL											.3	.35	.48	.4	.53
Within-country classification measured with sender DE-12																
DE12-within-ae	All											.81	.9	.94	.96	.97
DE12-within-ae	AE											.86	.95	.96	.98	.97
DE12-within-ae	NOT-AE											.75	.86	.92	.94	.96
DE12-within-de	All	.69	.76	.8	.8	.82	.67	.76	.82	.82	.85	.79	.85	.85	.88	.9
DE12-within-de	DE	.8	.74	.89	.85	.86	.83	.78	.81	.83	.84	.75	.84	.86	.9	.92
DE12-within-de	NOT-DE	.57	.79	.71	.74	.78	.51	.74	.83	.82	.85	.83	.85	.85	.87	.88
DE12-within-gr	All	.53	.56	.49	.59	.71	.65	.7	.78	.72	.72	.73	.8	.79	.85	.87
DE12-within-gr	GR	.6	.52	.5	.63	.68	.49	.74	.83	.68	.7	.57	.87	.87	.88	.91
DE12-within-gr	NOT-GR	.46	.59	.49	.55	.74	.81	.66	.72	.76	.73	.89	.73	.72	.82	.84
DE12-within-nl	All	.74	.81	.84	.86	.87	.66	.78	.77	.87	.86	.67	.76	.76	.76	.77
DE12-within-nl	NL	.65	.85	.85	.82	.86	.6	.78	.8	.87	.86	.61	.79	.78	.78	.81
DE12-within-nl	NOT-NL	.82	.77	.82	.9	.88	.72	.78	.73	.87	.86	.73	.73	.75	.73	.72
Two countries measured with sender GR-11																
GR11-2countries1	All						.59	.61	.66	.68	.68	.75	.84	.83	.87	.91
GR11-2countries1	DE						.85	.91	.7	.8	.53	.8	.86	.8	.84	.86
GR11-2countries1	NL						.32	.3	.62	.55	.83	.7	.81	.87	.9	.97

Continued on next page

Table C.1 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
GR11-2countries2	All											.88	.96	.96	.96	.98
GR11-2countries2	AE											.88	.99	.98	.97	.99
GR11-2countries2	DE											.88	.93	.93	.95	.98
GR11-2countries3	All											.68	.79	.86	.89	.95
GR11-2countries3	AE											.87	.92	.92	.91	.95
GR11-2countries3	NL											.48	.66	.8	.86	.95
Three countries measured with sender GR-11																
GR11-3countries	All											.62	.74	.82	.83	.9
GR11-3countries	AE											.81	.88	.93	.87	.95
GR11-3countries	DE											.88	.8	.8	.83	.89
GR11-3countries	NL											.18	.54	.74	.79	.88
Within-country classification measured with sender GR-11																
GR11-within-ae	All											.84	.9	.93	.93	.94
GR11-within-ae	AE											.88	.95	.97	.96	.94
GR11-within-ae	NOT-AE											.8	.86	.9	.9	.94
GR11-within-de	All						.53	.59	.62	.62	.7	.82	.87	.89	.9	.92
GR11-within-de	DE						.29	.87	.62	.66	.9	.86	.88	.89	.87	.9
GR11-within-de	NOT-DE						.76	.3	.63	.58	.51	.78	.85	.9	.93	.94
GR11-within-nl	All						.53	.6	.62	.66	.68	.64	.72	.79	.86	.88
GR11-within-nl	NL						.61	.3	.45	.71	.7	.7	.69	.8	.89	.91
GR11-within-nl	NOT-NL						.45	.89	.79	.62	.65	.58	.74	.79	.84	.85

C.2. Round 2 (United Arab Emirates)

Table C.2. Detailed classification results for the second round of measurements in the UAE.

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
Two cellular locations											
2mloc1-AE-2ALL	All	.5	.5	.52	.48	.51	.65	.85	.85	.85	.91
2mloc1-AE-2ALL	m-AE-A	.6	.54	.47	.42	.49	.85	.88	.87	.87	.92
2mloc1-AE-2ALL	m-AE-D	.4	.45	.57	.54	.53	.46	.81	.82	.84	.89
2mloc1-AE-22	All	.49	.49	.5	.52	.5	.63	.82	.87	.85	.9
2mloc1-AE-22	m-AE-A	.68	.51	.44	.51	.44	.79	.87	.91	.88	.93
2mloc1-AE-22	m-AE-D	.3	.47	.57	.52	.56	.47	.77	.82	.83	.87
2mloc2-AE-2ALL	All	.49	.51	.49	.52	.49	.73	.89	.89	.94	.92
2mloc2-AE-2ALL	m-AE-B	.36	.45	.52	.52	.54	.52	.86	.86	.91	.92
2mloc2-AE-2ALL	m-AE-A	.61	.57	.45	.52	.43	.94	.93	.93	.96	.93
2mloc3-AE-2ALL	All	.59	.57	.64	.65	.64	.82	.94	.95	.96	.96
2mloc3-AE-2ALL	m-AE-C	.63	.43	.59	.63	.64	.88	.95	.95	.97	.97
2mloc3-AE-2ALL	m-AE-A	.55	.72	.68	.68	.64	.76	.93	.95	.95	.96
2mloc4-AE-2ALL	All	.49	.55	.52	.52	.5	.65	.8	.81	.83	.88
2mloc4-AE-2ALL	m-AE-B	.34	.4	.54	.47	.51	.44	.79	.78	.8	.91
2mloc4-AE-2ALL	m-AE-D	.63	.71	.51	.56	.49	.85	.82	.84	.86	.84
2mloc5-AE-2ALL	All	.55	.56	.64	.67	.63	.71	.74	.75	.76	.8
2mloc5-AE-2ALL	m-AE-C	.54	.4	.62	.63	.56	.83	.88	.86	.8	.82
2mloc5-AE-2ALL	m-AE-D	.57	.73	.66	.71	.69	.59	.59	.64	.71	.79
2mloc6-AE-2ALL	All	.53	.56	.62	.68	.68	.83	.93	.91	.9	.92
2mloc6-AE-2ALL	m-AE-C	.62	.5	.58	.66	.6	.89	.96	.93	.92	.9
2mloc6-AE-2ALL	m-AE-B	.44	.62	.65	.7	.75	.76	.9	.89	.88	.93
Two cellular locations and any WiFi location											
2mlocw1-AE-2ALL	All	.33	.32	.37	.4	.4	.6	.74	.77	.78	.81
2mlocw1-AE-2ALL	m-AE-A	.6	.42	.43	.35	.32	.82	.82	.87	.83	.87
2mlocw1-AE-2ALL	m-AE-D	.2	.25	.25	.28	.35	.22	.62	.7	.71	.77
2mlocw1-AE-2ALL	wifi	.2	.29	.44	.58	.54	.76	.77	.76	.79	.78

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
2mlocw1-AE-22	All	.33	.36	.37	.38	.35	.52	.76	.8	.83	.85
2mlocw1-AE-22	m-AE-A	.33	.36	.54	.37	.36	.73	.88	.85	.89	.87
2mlocw1-AE-22	m-AE-D	.34	.33	.22	.26	.31	.02	.66	.69	.74	.84
2mlocw1-AE-22	wifi	.32	.37	.35	.53	.38	.83	.74	.86	.86	.84
2mlocw2-AE-2ALL	All	.33	.33	.35	.39	.4	.6	.79	.82	.82	.82
2mlocw2-AE-2ALL	m-AE-B	.32	.38	.3	.32	.36	.17	.63	.72	.71	.78
2mlocw2-AE-2ALL	m-AE-A	.39	.31	.35	.35	.4	.88	.93	.9	.92	.89
2mlocw2-AE-2ALL	wifi	.26	.31	.41	.5	.46	.76	.81	.83	.82	.79
2mlocw3-AE-2ALL	All	.36	.37	.43	.48	.44	.73	.82	.86	.86	.88
2mlocw3-AE-2ALL	m-AE-C	.53	.43	.48	.49	.49	.73	.87	.84	.85	.87
2mlocw3-AE-2ALL	m-AE-A	.29	.42	.33	.46	.35	.74	.9	.93	.94	.93
2mlocw3-AE-2ALL	wifi	.26	.25	.48	.5	.48	.71	.7	.81	.81	.85
2mlocw4-AE-2ALL	All	.33	.35	.36	.39	.4	.53	.67	.7	.67	.75
2mlocw4-AE-2ALL	m-AE-B	.42	.33	.43	.36	.35	.19	.66	.69	.68	.81
2mlocw4-AE-2ALL	m-AE-D	.41	.31	.25	.28	.38	.64	.72	.73	.73	.72
2mlocw4-AE-2ALL	wifi	.16	.41	.39	.52	.47	.75	.62	.67	.61	.73
2mlocw5-AE-2ALL	All	.36	.38	.45	.49	.46	.64	.65	.7	.7	.74
2mlocw5-AE-2ALL	m-AE-C	.69	.5	.54	.55	.47	.81	.83	.75	.81	.76
2mlocw5-AE-2ALL	m-AE-D	.26	.27	.29	.38	.4	.45	.5	.59	.52	.74
2mlocw5-AE-2ALL	wifi	.13	.38	.51	.55	.5	.66	.64	.77	.76	.73
2mlocw6-AE-2ALL	All	.33	.41	.45	.47	.48	.62	.71	.73	.74	.77
2mlocw6-AE-2ALL	m-AE-C	.24	.46	.54	.56	.45	.84	.85	.84	.83	.81
2mlocw6-AE-2ALL	m-AE-B	.25	.51	.42	.41	.56	.46	.76	.77	.76	.85
2mlocw6-AE-2ALL	wifi	.51	.26	.38	.45	.41	.56	.53	.59	.63	.66
Two WiFi locations											
2wloc1-AE-2ALL	All	.56	.59	.62	.65	.65	.53	.8	.78	.86	.79
2wloc1-AE-2ALL	w-AE-A	.49	.59	.65	.7	.66	.26	.79	.79	.88	.75
2wloc1-AE-2ALL	w-AE-D	.62	.58	.6	.61	.64	.79	.81	.77	.84	.82
2wloc1-AE-22	All	.54	.6	.65	.64	.65	.52	.83	.8	.83	.82
2wloc1-AE-22	w-AE-A	.64	.5	.73	.63	.61	.36	.85	.77	.86	.86
2wloc1-AE-22	w-AE-D	.45	.69	.57	.65	.69	.68	.81	.83	.8	.77
2wloc2-AE-2ALL	All	.58	.52	.49	.54	.49	.69	.87	.91	.91	.91
2wloc2-AE-2ALL	w-AE-B	.63	.43	.57	.6	.61	.63	.87	.91	.89	.87
2wloc2-AE-2ALL	w-AE-A	.53	.61	.41	.48	.38	.76	.86	.91	.92	.94

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
2wloc3-AE-2ALL	All	.53	.62	.61	.66	.67	.76	.92	.92	.93	.93
2wloc3-AE-2ALL	w-AE-C	.89	.41	.42	.52	.6	.83	.9	.91	.91	.92
2wloc3-AE-2ALL	w-AE-A	.16	.84	.81	.8	.74	.69	.94	.93	.94	.94
2wloc4-AE-2ALL	All	.51	.59	.63	.69	.67	.68	.75	.78	.84	.89
2wloc4-AE-2ALL	w-AE-B	.48	.5	.67	.78	.65	.76	.79	.8	.81	.9
2wloc4-AE-2ALL	w-AE-D	.54	.68	.6	.6	.7	.59	.7	.76	.87	.88
2wloc5-AE-2ALL	All	.59	.57	.67	.69	.73	.75	.88	.88	.87	.92
2wloc5-AE-2ALL	w-AE-C	.76	.6	.6	.63	.65	.78	.87	.84	.81	.9
2wloc5-AE-2ALL	w-AE-D	.43	.54	.74	.74	.82	.72	.9	.92	.93	.95
2wloc6-AE-2ALL	All	.58	.65	.65	.66	.67	.8	.93	.88	.9	.93
2wloc6-AE-2ALL	w-AE-C	.42	.58	.56	.6	.58	.74	.92	.89	.85	.91
2wloc6-AE-2ALL	w-AE-B	.75	.73	.75	.73	.77	.86	.94	.87	.94	.95
Two WiFi locations and any cellular location											
2wlocm1-AE-2ALL	All	.36	.38	.45	.46	.45	.44	.69	.66	.69	.7
2wlocm1-AE-2ALL	mobile	.1	.12	.32	.38	.27	.44	.68	.69	.73	.74
2wlocm1-AE-2ALL	w-AE-A	.47	.53	.54	.62	.57	.37	.73	.68	.76	.64
2wlocm1-AE-2ALL	w-AE-D	.53	.49	.49	.38	.53	.52	.66	.61	.57	.71
2wlocm1-AE-22	All	.36	.4	.45	.45	.44	.52	.75	.73	.75	.76
2wlocm1-AE-22	mobile	.29	.25	.34	.27	.38	.68	.75	.85	.8	.77
2wlocm1-AE-22	w-AE-A	.53	.59	.6	.67	.56	.57	.75	.69	.77	.75
2wlocm1-AE-22	w-AE-D	.27	.35	.4	.4	.37	.31	.75	.65	.66	.76
2wlocm2-AE-2ALL	All	.37	.38	.44	.45	.45	.55	.7	.76	.76	.8
2wlocm2-AE-2ALL	mobile	.02	.4	.53	.51	.53	.53	.45	.77	.75	.75
2wlocm2-AE-2ALL	w-AE-B	.54	.44	.35	.53	.37	.68	.85	.74	.78	.83
2wlocm2-AE-2ALL	w-AE-A	.54	.3	.45	.32	.44	.43	.79	.78	.74	.81
2wlocm3-AE-2ALL	All	.37	.37	.51	.5	.55	.62	.77	.81	.82	.84
2wlocm3-AE-2ALL	mobile	.34	.34	.46	.51	.56	.49	.73	.82	.76	.77
2wlocm3-AE-2ALL	w-AE-C	.68	.35	.36	.43	.54	.85	.79	.77	.83	.87
2wlocm3-AE-2ALL	w-AE-A	.09	.44	.71	.55	.55	.53	.78	.84	.86	.88
2wlocm4-AE-2ALL	All	.35	.39	.44	.47	.47	.55	.69	.72	.75	.82
2wlocm4-AE-2ALL	mobile	.81	.11	.35	.31	.37	.47	.74	.76	.76	.86
2wlocm4-AE-2ALL	w-AE-B	.08	.45	.67	.69	.56	.64	.65	.69	.74	.85
2wlocm4-AE-2ALL	w-AE-D	.16	.62	.3	.39	.48	.54	.69	.71	.76	.77

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
2wlocm5-AE-2ALL	All	.38	.39	.44	.52	.55	.57	.76	.76	.79	.79
2wlocm5-AE-2ALL	mobile	.12	.3	.35	.45	.57	.42	.72	.74	.83	.79
2wlocm5-AE-2ALL	w-AE-C	.76	.42	.58	.6	.68	.78	.81	.76	.78	.74
2wlocm5-AE-2ALL	w-AE-D	.26	.46	.4	.5	.4	.53	.75	.77	.75	.82
2wlocm6-AE-2ALL	All	.38	.45	.49	.55	.54	.7	.74	.75	.8	.8
2wlocm6-AE-2ALL	mobile	.04	.38	.54	.54	.55	.76	.66	.71	.77	.79
2wlocm6-AE-2ALL	w-AE-C	.55	.41	.33	.48	.5	.66	.79	.79	.81	.77
2wlocm6-AE-2ALL	w-AE-B	.55	.55	.58	.63	.57	.7	.77	.73	.83	.83
Three cellular locations											
3mloc1-AE-2ALL	All	.33	.37	.34	.35	.35	.53	.76	.77	.77	.81
3mloc1-AE-2ALL	m-AE-B	.48	.36	.36	.32	.25	.46	.77	.78	.74	.85
3mloc1-AE-2ALL	m-AE-A	.11	.11	.31	.28	.4	.71	.86	.85	.89	.85
3mloc1-AE-2ALL	m-AE-D	.41	.64	.34	.44	.39	.41	.65	.69	.68	.72
3mloc2-AE-2ALL	All	.37	.37	.43	.44	.43	.57	.75	.74	.76	.78
3mloc2-AE-2ALL	m-AE-C	.56	.45	.57	.59	.52	.85	.87	.81	.8	.76
3mloc2-AE-2ALL	m-AE-A	.45	.42	.37	.31	.44	.71	.88	.9	.89	.92
3mloc2-AE-2ALL	m-AE-D	.11	.25	.37	.43	.32	.15	.5	.5	.59	.66
3mloc3-AE-2ALL	All	.37	.37	.43	.42	.44	.68	.87	.87	.87	.88
3mloc3-AE-2ALL	m-AE-C	.53	.46	.61	.62	.58	.81	.92	.9	.92	.9
3mloc3-AE-2ALL	m-AE-B	.06	.45	.23	.32	.37	.46	.79	.8	.82	.84
3mloc3-AE-2ALL	m-AE-A	.51	.2	.45	.33	.36	.77	.9	.89	.87	.9
3mloc4-AE-2ALL	All	.36	.39	.43	.47	.44	.59	.7	.71	.7	.76
3mloc4-AE-2ALL	m-AE-C	.57	.43	.59	.59	.49	.86	.84	.88	.82	.75
3mloc4-AE-2ALL	m-AE-B	.21	.35	.3	.42	.44	.45	.78	.8	.75	.87
3mloc4-AE-2ALL	m-AE-D	.29	.38	.41	.42	.39	.44	.48	.46	.51	.67
Three cellular locations and any WiFi location											
3mlocw1-AE-2ALL	All	.24	.26	.32	.28	.29	.44	.66	.69	.72	.72
3mlocw1-AE-2ALL	m-AE-B	.15	.34	.25	.17	.22	.24	.66	.69	.71	.77
3mlocw1-AE-2ALL	m-AE-A	.21	.02	.27	.24	.25	.69	.78	.85	.86	.83
3mlocw1-AE-2ALL	m-AE-D	.16	.4	.27	.24	.27	.23	.45	.5	.6	.64
3mlocw1-AE-2ALL	wifi	.46	.28	.48	.46	.42	.61	.75	.7	.71	.66

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
3mlocw2-AE-2ALL	All	.27	.3	.33	.37	.38	.51	.64	.68	.68	.73
3mlocw2-AE-2ALL	m-AE-C	.6	.42	.44	.47	.46	.74	.78	.73	.76	.75
3mlocw2-AE-2ALL	m-AE-A	.43	.26	.27	.32	.31	.74	.81	.85	.83	.84
3mlocw2-AE-2ALL	m-AE-D	.04	.24	.26	.24	.33	.0	.4	.49	.42	.65
3mlocw2-AE-2ALL	wifi	.01	.26	.36	.46	.42	.56	.55	.64	.71	.7
3mlocw3-AE-2ALL	All	.27	.28	.33	.38	.34	.53	.73	.76	.78	.78
3mlocw3-AE-2ALL	m-AE-C	.33	.41	.55	.51	.42	.77	.83	.84	.81	.81
3mlocw3-AE-2ALL	m-AE-B	.04	.4	.28	.25	.33	.17	.66	.72	.76	.79
3mlocw3-AE-2ALL	m-AE-A	.7	.12	.22	.21	.21	.73	.89	.87	.91	.86
3mlocw3-AE-2ALL	wifi	.0	.18	.26	.54	.41	.44	.52	.63	.62	.66
3mlocw4-AE-2ALL	All	.25	.31	.32	.36	.37	.47	.58	.62	.62	.67
3mlocw4-AE-2ALL	m-AE-C	.39	.41	.5	.55	.43	.81	.8	.76	.6	.71
3mlocw4-AE-2ALL	m-AE-B	.19	.45	.18	.17	.32	.25	.66	.73	.74	.86
3mlocw4-AE-2ALL	m-AE-D	.2	.31	.29	.26	.35	.35	.4	.4	.5	.55
3mlocw4-AE-2ALL	wifi	.23	.07	.33	.48	.36	.46	.46	.59	.65	.57
Three WiFi locations											
3wloc1-AE-2ALL	All	.37	.4	.44	.46	.45	.45	.69	.7	.74	.75
3wloc1-AE-2ALL	w-AE-B	.45	.48	.42	.5	.35	.57	.74	.69	.76	.88
3wloc1-AE-2ALL	w-AE-A	.54	.2	.35	.3	.33	.28	.76	.79	.81	.69
3wloc1-AE-2ALL	w-AE-D	.14	.52	.56	.59	.66	.49	.58	.64	.65	.69
3wloc2-AE-2ALL	All	.37	.44	.48	.53	.57	.54	.78	.76	.78	.75
3wloc2-AE-2ALL	w-AE-C	.86	.39	.37	.46	.55	.81	.88	.85	.79	.82
3wloc2-AE-2ALL	w-AE-A	.0	.61	.46	.63	.53	.49	.76	.75	.8	.66
3wloc2-AE-2ALL	w-AE-D	.24	.32	.62	.49	.62	.33	.7	.68	.75	.77
3wloc3-AE-2ALL	All	.39	.44	.42	.46	.47	.61	.82	.82	.86	.9
3wloc3-AE-2ALL	w-AE-C	.62	.38	.41	.47	.53	.72	.83	.8	.84	.86
3wloc3-AE-2ALL	w-AE-B	.54	.63	.49	.45	.43	.6	.79	.78	.82	.87
3wloc3-AE-2ALL	w-AE-A	.02	.32	.36	.46	.45	.5	.84	.89	.93	.95
3wloc4-AE-2ALL	All	.39	.46	.54	.53	.54	.59	.73	.8	.81	.85
3wloc4-AE-2ALL	w-AE-C	.61	.43	.35	.51	.54	.68	.83	.9	.83	.86
3wloc4-AE-2ALL	w-AE-B	.37	.51	.69	.57	.47	.57	.63	.66	.76	.85
3wloc4-AE-2ALL	w-AE-D	.18	.42	.57	.51	.62	.51	.72	.83	.86	.85

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
Three WiFi locations and any cellular location											
3wlocm1-AE-2ALL	All	.27	.29	.32	.37	.35	.46	.61	.64	.67	.71
3wlocm1-AE-2ALL	mobile	.08	.13	.42	.32	.31	.6	.48	.69	.76	.72
3wlocm1-AE-2ALL	w-AE-B	.3	.36	.35	.5	.35	.66	.64	.65	.7	.81
3wlocm1-AE-2ALL	w-AE-A	.6	.23	.3	.29	.33	.38	.72	.69	.8	.65
3wlocm1-AE-2ALL	w-AE-D	.12	.43	.2	.38	.41	.2	.61	.55	.42	.67
3wlocm2-AE-2ALL	All	.29	.32	.4	.44	.43	.41	.68	.67	.67	.71
3wlocm2-AE-2ALL	mobile	.13	.27	.31	.38	.22	.53	.6	.67	.66	.66
3wlocm2-AE-2ALL	w-AE-C	.71	.36	.33	.41	.52	.78	.84	.75	.77	.79
3wlocm2-AE-2ALL	w-AE-A	.0	.5	.59	.58	.54	.23	.63	.69	.62	.68
3wlocm2-AE-2ALL	w-AE-D	.3	.14	.35	.38	.42	.09	.66	.56	.63	.72
3wlocm3-AE-2ALL	All	.3	.35	.39	.4	.39	.51	.71	.74	.77	.8
3wlocm3-AE-2ALL	mobile	.13	.3	.45	.44	.38	.55	.51	.62	.64	.69
3wlocm3-AE-2ALL	w-AE-C	.42	.37	.32	.4	.54	.69	.75	.83	.83	.77
3wlocm3-AE-2ALL	w-AE-B	.55	.5	.46	.47	.32	.57	.79	.71	.77	.86
3wlocm3-AE-2ALL	w-AE-A	.1	.21	.34	.27	.3	.22	.77	.78	.86	.87
3wlocm4-AE-2ALL	All	.29	.34	.39	.42	.43	.55	.66	.69	.74	.76
3wlocm4-AE-2ALL	mobile	.27	.1	.38	.32	.29	.53	.54	.63	.68	.73
3wlocm4-AE-2ALL	w-AE-C	.43	.42	.36	.4	.56	.74	.79	.76	.8	.76
3wlocm4-AE-2ALL	w-AE-B	.29	.51	.58	.6	.41	.61	.66	.64	.74	.79
3wlocm4-AE-2ALL	w-AE-D	.18	.34	.22	.34	.44	.32	.65	.71	.73	.77
Four cellular locations and any WiFi location											
4mlocw-AE-2ALL	All	.22	.24	.26	.31	.27	.44	.61	.63	.63	.7
4mlocw-AE-2ALL	m-AE-C	.38	.35	.47	.5	.44	.8	.77	.76	.73	.7
4mlocw-AE-2ALL	m-AE-B	.0	.31	.18	.2	.21	.2	.66	.69	.69	.79
4mlocw-AE-2ALL	m-AE-A	.48	.18	.14	.2	.16	.75	.83	.88	.85	.82
4mlocw-AE-2ALL	m-AE-D	.01	.22	.18	.21	.24	.01	.28	.28	.29	.54
4mlocw-AE-2ALL	wifi	.24	.12	.31	.45	.31	.44	.52	.56	.58	.66
Four WiFi locations											
4wloc-AE-2ALL	All	.29	.36	.38	.4	.43	.46	.69	.69	.74	.75
4wloc-AE-2ALL	w-AE-C	.59	.39	.38	.41	.49	.71	.83	.85	.86	.8
4wloc-AE-2ALL	w-AE-B	.39	.47	.34	.56	.34	.55	.63	.69	.76	.84
4wloc-AE-2ALL	w-AE-A	.0	.28	.32	.2	.27	.34	.74	.77	.78	.7
4wloc-AE-2ALL	w-AE-D	.19	.3	.48	.41	.61	.24	.57	.47	.58	.65

Continued on next page

Table C.2 – continued from previous page

Classification	**Rec.**	**Signal**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5
Four WiFi locations and any cellular location											
4wlocm-AE-2ALL	All	.23	.29	.32	.32	.34	.45	.6	.66	.65	.67
4wlocm-AE-2ALL	mobile	.02	.26	.25	.26	.29	.49	.49	.71	.65	.56
4wlocm-AE-2ALL	w-AE-C	.45	.37	.34	.4	.47	.71	.72	.73	.78	.74
4wlocm-AE-2ALL	w-AE-B	.41	.35	.35	.43	.21	.66	.6	.64	.71	.81
4wlocm-AE-2ALL	w-AE-A	.02	.23	.33	.17	.34	.13	.7	.69	.62	.6
4wlocm-AE-2ALL	w-AE-D	.25	.22	.34	.31	.38	.24	.5	.55	.46	.65
All possible locations (WiFi and cellular, cross-receiver)											
all-AE-2ALL	All	.15	.19	.2	.25	.25	.31	.52	.58	.59	.65
all-AE-2ALL	m-AE-C	.22	.28	.26	.39	.24	.59	.37	.63	.58	.65
all-AE-2ALL	m-AE-B	.0	.08	.1	.14	.19	.01	.48	.55	.5	.7
all-AE-2ALL	m-AE-A	.0	.16	.08	.11	.1	.66	.77	.82	.88	.77
all-AE-2ALL	m-AE-D	.09	.03	.13	.14	.16	.02	.33	.34	.29	.37
all-AE-2ALL	w-AE-C	.4	.32	.35	.37	.46	.69	.77	.76	.75	.75
all-AE-2ALL	w-AE-B	.27	.31	.28	.39	.32	.52	.38	.39	.63	.77
all-AE-2ALL	w-AE-A	.01	.09	.29	.22	.21	.0	.57	.69	.59	.61
all-AE-2ALL	w-AE-D	.21	.27	.09	.27	.29	.0	.52	.48	.51	.61
All possible locations (WiFi and cellular, single receiver)											
all-AE-22	All	.26	.28	.32	.34	.34	.4	.68	.71	.69	.71
all-AE-22	m-AE-A	.67	.28	.26	.29	.25	.78	.84	.81	.84	.84
all-AE-22	m-AE-D	.1	.14	.25	.21	.25	.0	.65	.64	.59	.7
all-AE-22	w-AE-A	.14	.48	.57	.59	.56	.42	.56	.73	.7	.64
all-AE-22	w-AE-D	.14	.25	.21	.29	.31	.38	.68	.66	.64	.66
Network connection (WiFi vs. cellular) of receiver devices											
network-AE-2ALL	All	.5	.51	.6	.62	.62	.7	.77	.84	.86	.87
network-AE-2ALL	mobile	.6	.6	.54	.54	.55	.74	.76	.84	.86	.89
network-AE-2ALL	wifi	.4	.42	.65	.7	.69	.66	.77	.85	.86	.84
network-AE-22	All	.51	.52	.56	.54	.54	.75	.85	.89	.91	.91
network-AE-22	mobile	.76	.55	.54	.48	.51	.66	.89	.89	.94	.95
network-AE-22	wifi	.26	.48	.58	.6	.58	.85	.82	.89	.88	.88
network-AE-23	All	.48	.57	.62	.64	.61	.82	.78	.79	.85	.89
network-AE-23	mobile	.24	.51	.56	.58	.52	.73	.83	.8	.86	.92
network-AE-23	wifi	.71	.62	.68	.69	.71	.91	.72	.79	.85	.86

Continued on next page

Table C.2 – continued from previous page

Classification	Rec.	Signal					Whatsapp				
		1	2	3	4	5	1	2	3	4	5
network-AE-24	All	.52	.6	.63	.73	.77	.88	.89	.9	.89	.9
network-AE-24	mobile	.14	.73	.61	.7	.76	.89	.92	.92	.88	.93
network-AE-24	wifi	.9	.48	.64	.77	.78	.87	.86	.87	.9	.87

C.3. Round 2 (Germany)

Table C.3. Detailed classification results for the second round of measurements in Germany.

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Two cellular locations																
2mloc1-DE-2ALL	All	.55	.6	.64	.66	.73	.74	.87	.85	.85	.85	.79	.83	.85	.86	.88
2mloc1-DE-2ALL	m-DE-A	.69	.9	.81	.82	.75	.56	.93	.9	.91	.91	.59	.73	.78	.82	.89
2mloc1-DE-2ALL	m-DE-B	.42	.31	.46	.5	.71	.93	.81	.79	.8	.79	1.0	.93	.91	.89	.88
Two cellular locations and any WiFi location																
2mlocw1-DE-2ALL	All	.5	.57	.61	.66	.75	.55	.71	.63	.68	.68	.61	.73	.74	.77	.81
2mlocw1-DE-2ALL	m-DE-A	.72	.65	.69	.63	.74	.57	.73	.68	.75	.66	.58	.68	.74	.78	.82
2mlocw1-DE-2ALL	m-DE-B	.06	.29	.37	.51	.66	.61	.77	.66	.74	.82	.64	.72	.7	.71	.77
2mlocw1-DE-2ALL	wifi	.74	.75	.78	.83	.84	.46	.61	.56	.55	.56	.62	.79	.79	.82	.85
Two WiFi locations																
2wloc1-DE-23	All	.6	.69	.73	.79	.8	.81	.85	.86	.86	.82	.69	.7	.73	.75	.74
2wloc1-DE-23	w-DE-A	.47	.81	.81	.87	.84	.89	.91	.92	.9	.88	.8	.8	.88	.82	.82
2wloc1-DE-23	w-DE-B	.73	.57	.65	.71	.75	.73	.79	.81	.81	.75	.57	.59	.58	.69	.66
2wloc1-DE-2ALL	All	.7	.73	.76	.76	.79	.71	.76	.8	.8	.81	.7	.76	.79	.81	.81
2wloc1-DE-2ALL	w-DE-A	.43	.52	.57	.66	.73	.6	.63	.7	.73	.72	.49	.58	.62	.73	.74
2wloc1-DE-2ALL	w-DE-B	.96	.95	.95	.87	.84	.81	.89	.9	.87	.89	.91	.94	.95	.89	.89
2wloc1-DE-22	All	.53	.55	.59	.64	.67	.63	.67	.74	.71	.75	.53	.63	.64	.66	.65
2wloc1-DE-22	w-DE-A	.83	.71	.72	.78	.72	.43	.47	.65	.74	.65	.98	.69	.65	.59	.57
2wloc1-DE-22	w-DE-B	.23	.38	.47	.49	.63	.83	.87	.83	.67	.84	.08	.56	.62	.73	.72

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2wloc1-DE-24	All	.79	.86	.88	.92	.93	.84	.88	.87	.88	.88	.85	.89	.92	.93	.95
2wloc1-DE-24	w-DE-A	.63	.79	.82	.89	.91	.71	.81	.81	.82	.85	.73	.81	.92	.94	.96
2wloc1-DE-24	w-DE-B	.95	.93	.93	.95	.95	.96	.95	.93	.93	.92	.96	.98	.92	.93	.94
2wloc2-DE-2ALL	All	.77	.79	.78	.8	.81	.7	.77	.81	.85	.85	.79	.79	.82	.81	.83
2wloc2-DE-2ALL	w-DE-C	.92	.95	.93	.95	.86	.92	.87	.91	.92	.92	.96	.95	.98	.94	.94
2wloc2-DE-2ALL	w-DE-A	.61	.64	.64	.65	.76	.48	.68	.72	.78	.78	.61	.63	.66	.68	.72
2wloc2-DE-22	All	.52	.57	.6	.61	.61	.65	.68	.86	.86	.88	.5	.59	.62	.64	.68
2wloc2-DE-22	w-DE-C	.6	.73	.6	.59	.64	.58	.81	.84	.87	.92	.22	.42	.59	.64	.73
2wloc2-DE-22	w-DE-A	.44	.42	.6	.62	.59	.72	.55	.89	.85	.83	.78	.76	.65	.64	.62
2wloc3-DE-23	All	.51	.57	.61	.69	.71	.9	.92	.92	.91	.92	.6	.62	.63	.67	.8
2wloc3-DE-23	w-DE-D	.57	.27	.56	.59	.56	.88	.91	.89	.89	.88	.45	.5	.52	.67	.77
2wloc3-DE-23	w-DE-A	.46	.87	.66	.8	.86	.92	.94	.95	.94	.95	.75	.74	.74	.67	.84
2wloc3-DE-2ALL	All	.77	.77	.79	.79	.83	.85	.87	.89	.89	.9	.75	.79	.79	.83	.83
2wloc3-DE-2ALL	w-DE-D	.92	.94	.94	.9	.91	.82	.85	.87	.87	.88	.93	.95	.92	.92	.84
2wloc3-DE-2ALL	w-DE-A	.63	.61	.63	.68	.76	.87	.9	.9	.91	.92	.57	.63	.66	.73	.81
2wloc4-DE-2ALL	All	.8	.82	.9	.89	.91	.7	.87	.87	.83	.89	.8	.92	.92	.89	.9
2wloc4-DE-2ALL	w-DE-E	.83	.87	.92	.9	.94	.7	.87	.88	.8	.89	.8	.94	.93	.89	.91
2wloc4-DE-2ALL	w-DE-A	.76	.77	.88	.89	.88	.71	.88	.86	.86	.89	.8	.91	.9	.88	.89
2wloc4-DE-24	All	.76	.83	.88	.89	.88	.77	.88	.89	.9	.83	.87	.88	.9	.9	.89
2wloc4-DE-24	w-DE-E	.89	.94	.97	.94	.91	.86	.95	.94	.94	.89	.99	.97	.9	.89	.91
2wloc4-DE-24	w-DE-A	.63	.73	.79	.84	.86	.68	.81	.84	.86	.77	.76	.79	.9	.9	.86
2wloc5-DE-2ALL	All	.6	.64	.64	.7	.76	.65	.66	.69	.69	.69	.59	.58	.6	.57	.61
2wloc5-DE-2ALL	w-DE-C	.9	.84	.76	.83	.82	.7	.85	.77	.72	.75	.92	.87	.92	.81	.71
2wloc5-DE-2ALL	w-DE-B	.3	.45	.53	.58	.71	.6	.48	.61	.65	.64	.25	.3	.28	.33	.51
2wloc5-DE-22	All	.52	.5	.53	.56	.6	.66	.69	.68	.67	.68	.52	.52	.53	.53	.62
2wloc5-DE-22	w-DE-C	.76	.88	.88	.74	.76	.58	.53	.57	.52	.61	.98	.53	.56	.56	.68
2wloc5-DE-22	w-DE-B	.28	.12	.18	.37	.43	.74	.84	.8	.82	.75	.05	.52	.5	.49	.57
2wloc6-DE-23	All	.55	.58	.63	.64	.66	.77	.79	.8	.81	.83	.65	.67	.65	.67	.69
2wloc6-DE-23	w-DE-D	.45	.69	.62	.74	.7	.76	.87	.8	.78	.8	.92	.87	.85	.78	.68
2wloc6-DE-23	w-DE-B	.66	.47	.65	.54	.62	.78	.71	.79	.84	.86	.38	.46	.45	.56	.71
2wloc6-DE-2ALL	All	.63	.63	.64	.63	.74	.82	.88	.87	.87	.89	.57	.56	.57	.59	.71
2wloc6-DE-2ALL	w-DE-D	.88	.81	.77	.74	.83	.81	.87	.85	.85	.85	.87	.84	.87	.73	.66
2wloc6-DE-2ALL	w-DE-B	.38	.45	.51	.52	.64	.83	.89	.89	.88	.93	.28	.29	.27	.44	.75

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2wloc7-DE-2ALL	All	.73	.79	.77	.87	.9	.81	.87	.88	.81	.83	.8	.88	.89	.91	.89
2wloc7-DE-2ALL	w-DE-E	.83	.85	.86	.9	.92	.82	.93	.94	.88	.85	.76	.93	.93	.95	.92
2wloc7-DE-2ALL	w-DE-B	.63	.73	.68	.84	.88	.79	.81	.81	.75	.8	.84	.83	.85	.87	.85
2wloc7-DE-24	All	.53	.44	.57	.73	.79	.78	.84	.83	.74	.79	.52	.5	.51	.56	.58
2wloc7-DE-24	w-DE-E	.52	.36	.53	.84	.83	.64	.71	.79	.85	.91	.62	.57	.59	.51	.57
2wloc7-DE-24	w-DE-B	.55	.53	.61	.62	.75	.92	.96	.86	.63	.67	.41	.43	.43	.61	.59
2wloc8-DE-2ALL	All	.5	.53	.56	.64	.85	.91	.96	.95	.96	.95	.51	.56	.56	.58	.71
2wloc8-DE-2ALL	w-DE-D	.2	.19	.61	.51	.89	.88	.95	.97	.98	.98	.21	.66	.48	.42	.66
2wloc8-DE-2ALL	w-DE-C	.81	.87	.51	.76	.8	.95	.96	.93	.95	.93	.81	.46	.64	.74	.76
2wloc9-DE-2ALL	All	.87	.86	.92	.95	.96	.79	.87	.89	.85	.83	.84	.96	.96	.99	.99
2wloc9-DE-2ALL	w-DE-C	.93	.85	.94	.96	.94	.96	.92	.93	.84	.84	.92	.97	.95	.98	.98
2wloc9-DE-2ALL	w-DE-E	.81	.87	.9	.95	.98	.61	.82	.84	.85	.82	.76	.95	.97	.99	.99
2wloc10-DE-2ALL	All	.86	.89	.89	.94	.95	.84	.93	.97	.94	.97	.84	.97	.98	.98	.99
2wloc10-DE-2ALL	w-DE-D	.9	.93	.93	.96	.94	.75	.9	.95	.91	.98	.94	.99	.99	.99	.99
2wloc10-DE-2ALL	w-DE-E	.82	.86	.86	.92	.96	.92	.96	.99	.96	.96	.75	.94	.98	.98	.99
Two WiFi locations and any cellular location																
2wlocm1-DE-23	All	.58	.64	.67	.75	.78	.65	.76	.68	.68	.64	.61	.63	.66	.72	.74
2wlocm1-DE-23	mobile	.69	.74	.73	.76	.85	.74	.68	.64	.53	.58	.84	.86	.89	.81	.79
2wlocm1-DE-23	w-DE-A	.36	.63	.66	.78	.79	.84	.85	.65	.73	.67	.8	.78	.81	.83	.83
2wlocm1-DE-23	w-DE-B	.69	.56	.62	.7	.7	.37	.73	.75	.77	.66	.19	.24	.29	.51	.6
2wlocm1-DE-2ALL	All	.58	.66	.68	.73	.79	.54	.65	.74	.75	.77	.59	.7	.72	.76	.78
2wlocm1-DE-2ALL	mobile	.67	.71	.72	.77	.86	.47	.68	.78	.8	.81	.55	.72	.78	.79	.86
2wlocm1-DE-2ALL	w-DE-A	.41	.51	.59	.7	.72	.47	.55	.64	.65	.67	.44	.55	.58	.66	.65
2wlocm1-DE-2ALL	w-DE-B	.66	.75	.73	.72	.78	.68	.73	.79	.78	.82	.79	.82	.81	.82	.82
2wlocm1-DE-22	All	.55	.57	.61	.61	.71	.52	.58	.66	.65	.7	.46	.61	.66	.7	.71
2wlocm1-DE-22	mobile	.69	.72	.78	.72	.89	.46	.65	.59	.65	.66	.79	.79	.84	.9	.92
2wlocm1-DE-22	w-DE-A	.6	.62	.62	.65	.69	.3	.34	.64	.59	.71	.41	.71	.53	.54	.66
2wlocm1-DE-22	w-DE-B	.36	.36	.42	.46	.55	.79	.75	.73	.73	.73	.19	.33	.6	.67	.54
2wlocm1-DE-24	All	.61	.76	.78	.89	.9	.58	.76	.82	.86	.85	.63	.78	.84	.87	.87
2wlocm1-DE-24	mobile	.76	.74	.74	.91	.92	.74	.68	.87	.87	.9	.73	.82	.8	.91	.86
2wlocm1-DE-24	w-DE-A	.64	.78	.78	.84	.89	.72	.81	.81	.85	.81	.71	.79	.86	.87	.87
2wlocm1-DE-24	w-DE-B	.43	.78	.82	.92	.89	.3	.8	.79	.85	.85	.45	.75	.87	.83	.87

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2wlocm2-DE-2ALL	All	.66	.71	.69	.71	.79	.5	.69	.75	.74	.79	.64	.73	.75	.76	.77
2wlocm2-DE-2ALL	mobile	.66	.72	.71	.77	.89	.29	.61	.69	.64	.75	.58	.75	.74	.83	.78
2wlocm2-DE-2ALL	w-DE-C	.89	.83	.84	.8	.82	.73	.82	.87	.84	.83	.94	.92	.95	.9	.94
2wlocm2-DE-2ALL	w-DE-A	.44	.58	.53	.56	.67	.48	.63	.7	.74	.78	.41	.52	.57	.54	.57
2wlocm2-DE-22	All	.57	.6	.6	.61	.68	.44	.62	.73	.73	.8	.45	.61	.61	.69	.72
2wlocm2-DE-22	mobile	.7	.72	.73	.79	.93	.48	.58	.6	.58	.71	.78	.75	.84	.89	.91
2wlocm2-DE-22	w-DE-C	.46	.58	.54	.51	.57	.17	.78	.77	.81	.84	.46	.52	.56	.6	.63
2wlocm2-DE-22	w-DE-A	.55	.51	.52	.52	.55	.66	.49	.81	.81	.85	.12	.55	.42	.58	.63
2wlocm3-DE-23	All	.54	.57	.62	.7	.75	.78	.82	.71	.7	.72	.64	.65	.68	.72	.81
2wlocm3-DE-23	mobile	.71	.72	.78	.81	.9	.69	.73	.58	.6	.62	.8	.87	.9	.93	.91
2wlocm3-DE-23	w-DE-D	.58	.48	.41	.62	.6	.8	.84	.87	.86	.89	.38	.44	.44	.58	.68
2wlocm3-DE-23	w-DE-A	.33	.52	.66	.67	.74	.86	.88	.68	.65	.64	.73	.64	.7	.65	.85
2wlocm3-DE-2ALL	All	.64	.69	.7	.74	.81	.63	.73	.77	.8	.81	.61	.72	.73	.75	.79
2wlocm3-DE-2ALL	mobile	.62	.72	.78	.81	.82	.59	.68	.78	.87	.82	.52	.72	.78	.82	.85
2wlocm3-DE-2ALL	w-DE-D	.86	.83	.82	.82	.87	.81	.76	.79	.81	.84	.87	.91	.88	.9	.83
2wlocm3-DE-2ALL	w-DE-A	.43	.53	.49	.59	.74	.49	.73	.75	.73	.75	.42	.54	.53	.54	.7
2wlocm4-DE-2ALL	All	.54	.66	.69	.76	.85	.56	.75	.81	.79	.75	.57	.7	.77	.79	.82
2wlocm4-DE-2ALL	mobile	.57	.57	.62	.83	.84	.72	.65	.78	.79	.76	.44	.59	.71	.73	.76
2wlocm4-DE-2ALL	w-DE-E	.39	.75	.8	.78	.88	.54	.85	.88	.82	.79	.8	.71	.81	.87	.89
2wlocm4-DE-2ALL	w-DE-A	.66	.66	.66	.67	.83	.43	.75	.78	.75	.72	.47	.8	.8	.78	.8
2wlocm4-DE-24	All	.58	.78	.81	.85	.85	.74	.85	.88	.85	.86	.63	.75	.82	.84	.78
2wlocm4-DE-24	mobile	.56	.74	.82	.9	.89	.9	.87	.93	.85	.89	.41	.64	.83	.85	.79
2wlocm4-DE-24	w-DE-E	.52	.8	.8	.87	.86	.6	.89	.89	.89	.86	.76	.76	.82	.79	.81
2wlocm4-DE-24	w-DE-A	.65	.79	.8	.77	.78	.72	.8	.82	.8	.83	.74	.85	.82	.89	.74
2wlocm5-DE-2ALL	All	.54	.6	.59	.67	.77	.47	.62	.63	.64	.67	.49	.58	.59	.62	.66
2wlocm5-DE-2ALL	mobile	.66	.66	.72	.73	.88	.35	.65	.65	.68	.76	.52	.76	.75	.85	.84
2wlocm5-DE-2ALL	w-DE-C	.9	.8	.69	.8	.84	.58	.6	.67	.69	.69	.89	.85	.88	.84	.62
2wlocm5-DE-2ALL	w-DE-B	.07	.33	.37	.47	.59	.49	.6	.56	.56	.57	.07	.13	.14	.16	.52
2wlocm5-DE-22	All	.53	.51	.54	.59	.66	.51	.67	.71	.7	.72	.47	.56	.61	.62	.69
2wlocm5-DE-22	mobile	.68	.71	.74	.77	.88	.47	.76	.86	.85	.9	.72	.77	.82	.87	.93
2wlocm5-DE-22	w-DE-C	.66	.61	.76	.43	.56	.29	.53	.6	.54	.67	.56	.49	.63	.62	.63
2wlocm5-DE-22	w-DE-B	.26	.2	.11	.56	.54	.77	.72	.68	.71	.59	.14	.41	.37	.35	.52

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2wlocm6-DE-23	All	.57	.6	.64	.66	.72	.63	.76	.77	.78	.75	.58	.61	.63	.67	.71
2wlocm6-DE-23	mobile	.71	.72	.78	.8	.9	.84	.93	.88	.96	.89	.82	.79	.89	.8	.84
2wlocm6-DE-23	w-DE-D	.34	.49	.55	.56	.66	.76	.79	.78	.76	.81	.9	.87	.9	.77	.64
2wlocm6-DE-23	w-DE-B	.66	.58	.59	.63	.6	.28	.56	.64	.62	.56	.02	.17	.09	.43	.64
2wlocm6-DE-2ALL	All	.52	.57	.6	.64	.71	.68	.73	.76	.79	.82	.49	.57	.59	.59	.71
2wlocm6-DE-2ALL	mobile	.64	.65	.71	.78	.83	.63	.66	.75	.83	.8	.57	.73	.81	.78	.79
2wlocm6-DE-2ALL	w-DE-D	.86	.78	.72	.62	.74	.77	.78	.81	.79	.85	.82	.84	.83	.72	.66
2wlocm6-DE-2ALL	w-DE-B	.07	.29	.36	.51	.55	.63	.75	.72	.76	.81	.08	.14	.15	.28	.69
2wlocm7-DE-2ALL	All	.5	.6	.65	.71	.78	.6	.73	.78	.75	.75	.56	.67	.71	.76	.76
2wlocm7-DE-2ALL	mobile	.41	.56	.58	.62	.73	.6	.72	.76	.69	.72	.18	.5	.6	.72	.75
2wlocm7-DE-2ALL	w-DE-E	.52	.72	.75	.89	.91	.51	.83	.89	.92	.8	.77	.8	.84	.87	.81
2wlocm7-DE-2ALL	w-DE-B	.58	.54	.63	.61	.7	.67	.63	.68	.64	.74	.74	.71	.69	.7	.74
2wlocm7-DE-24	All	.41	.56	.56	.73	.81	.56	.71	.79	.76	.74	.41	.51	.62	.62	.67
2wlocm7-DE-24	mobile	.49	.77	.79	.87	.9	.8	.69	.83	.89	.85	.38	.68	.78	.85	.84
2wlocm7-DE-24	w-DE-E	.43	.51	.59	.87	.86	.68	.72	.77	.73	.73	.48	.43	.44	.42	.51
2wlocm7-DE-24	w-DE-B	.32	.39	.31	.45	.67	.2	.72	.76	.66	.63	.37	.41	.64	.58	.65
2wlocm8-DE-2ALL	All	.53	.54	.59	.63	.84	.61	.77	.79	.8	.84	.5	.56	.61	.61	.74
2wlocm8-DE-2ALL	mobile	.67	.66	.74	.77	.9	.22	.56	.63	.62	.72	.57	.76	.79	.83	.83
2wlocm8-DE-2ALL	w-DE-D	.51	.47	.46	.63	.83	.78	.83	.87	.86	.92	.26	.28	.42	.24	.6
2wlocm8-DE-2ALL	w-DE-C	.41	.49	.58	.51	.8	.83	.9	.85	.9	.88	.67	.64	.62	.77	.78
2wlocm9-DE-2ALL	All	.61	.68	.74	.8	.9	.55	.74	.77	.74	.73	.63	.76	.81	.81	.84
2wlocm9-DE-2ALL	mobile	.38	.48	.48	.67	.87	.45	.65	.63	.64	.69	.18	.55	.69	.7	.75
2wlocm9-DE-2ALL	w-DE-C	.96	.79	.85	.84	.93	.64	.87	.84	.81	.72	.95	.94	.9	.92	.91
2wlocm9-DE-2ALL	w-DE-E	.48	.77	.89	.87	.91	.55	.71	.84	.78	.77	.75	.78	.84	.81	.87
2wlocm10-DE-2ALL	All	.61	.74	.72	.81	.87	.67	.78	.84	.81	.84	.58	.69	.76	.84	.8
2wlocm10-DE-2ALL	mobile	.53	.58	.54	.67	.78	.78	.71	.8	.75	.78	.18	.36	.6	.73	.67
2wlocm10-DE-2ALL	w-DE-D	.89	.85	.78	.82	.91	.66	.79	.81	.78	.88	.8	.9	.86	.93	.88
2wlocm10-DE-2ALL	w-DE-E	.42	.77	.82	.93	.91	.56	.85	.91	.89	.87	.76	.82	.8	.87	.86
Three WiFi locations																
3wloc1-DE-2ALL	All	.54	.57	.6	.61	.7	.54	.59	.63	.64	.67	.52	.58	.58	.59	.6
3wloc1-DE-2ALL	w-DE-C	.88	.95	.8	.84	.83	.63	.67	.63	.67	.72	.95	.93	.9	.82	.69
3wloc1-DE-2ALL	w-DE-A	.46	.49	.54	.48	.7	.46	.53	.61	.61	.71	.49	.58	.57	.67	.62
3wloc1-DE-2ALL	w-DE-B	.29	.26	.46	.51	.58	.54	.58	.67	.64	.59	.11	.22	.26	.28	.47

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
3wloc1-DE-22	All	.38	.4	.42	.43	.45	.5	.55	.61	.59	.62	.34	.42	.4	.43	.49
3wloc1-DE-22	w-DE-C	.37	.56	.5	.42	.45	.54	.57	.57	.58	.6	.19	.11	.3	.27	.44
3wloc1-DE-22	w-DE-A	.56	.48	.54	.53	.48	.22	.31	.67	.59	.64	.58	.61	.62	.56	.56
3wloc1-DE-22	w-DE-B	.21	.16	.23	.33	.43	.75	.76	.58	.6	.62	.25	.54	.29	.46	.47
3wloc2-DE-23	All	.4	.45	.48	.55	.59	.7	.75	.77	.8	.76	.49	.5	.53	.54	.63
3wloc2-DE-23	w-DE-D	.05	.15	.27	.4	.42	.73	.81	.79	.78	.79	.32	.45	.42	.57	.61
3wloc2-DE-23	w-DE-A	.53	.79	.56	.78	.78	.91	.84	.91	.93	.92	.74	.62	.74	.64	.8
3wloc2-DE-23	w-DE-B	.62	.41	.61	.47	.58	.47	.6	.61	.7	.57	.41	.42	.42	.42	.48
3wloc2-DE-2ALL	All	.51	.56	.61	.59	.64	.67	.71	.75	.78	.77	.52	.56	.54	.57	.67
3wloc2-DE-2ALL	w-DE-D	.91	.76	.86	.81	.77	.75	.77	.82	.83	.83	.86	.85	.77	.78	.76
3wloc2-DE-2ALL	w-DE-A	.43	.48	.54	.61	.6	.55	.53	.66	.69	.65	.56	.54	.56	.66	.64
3wloc2-DE-2ALL	w-DE-B	.19	.45	.43	.35	.56	.71	.83	.77	.83	.83	.14	.28	.29	.28	.6
3wloc3-DE-2ALL	All	.57	.62	.69	.75	.79	.64	.71	.76	.71	.78	.64	.74	.76	.75	.75
3wloc3-DE-2ALL	w-DE-E	.67	.76	.9	.93	.9	.71	.91	.89	.84	.87	.72	.93	.92	.89	.9
3wloc3-DE-2ALL	w-DE-A	.39	.46	.44	.6	.56	.46	.5	.56	.59	.61	.42	.54	.54	.54	.52
3wloc3-DE-2ALL	w-DE-B	.64	.64	.73	.74	.9	.75	.72	.83	.7	.85	.79	.74	.83	.81	.84
3wloc3-DE-24	All	.58	.6	.6	.7	.77	.69	.81	.81	.75	.78	.61	.62	.63	.65	.66
3wloc3-DE-24	w-DE-E	.24	.51	.48	.86	.84	.49	.74	.74	.81	.81	.7	.72	.65	.55	.59
3wloc3-DE-24	w-DE-A	.77	.79	.75	.73	.81	.67	.78	.79	.79	.82	.71	.78	.87	.87	.89
3wloc3-DE-24	w-DE-B	.73	.5	.58	.51	.66	.92	.9	.89	.65	.71	.41	.38	.36	.53	.51
3wloc4-DE-2ALL	All	.52	.54	.56	.61	.77	.7	.77	.81	.82	.84	.52	.57	.58	.61	.68
3wloc4-DE-2ALL	w-DE-D	.09	.36	.56	.51	.86	.8	.84	.88	.85	.86	.09	.26	.55	.54	.64
3wloc4-DE-2ALL	w-DE-C	.88	.68	.53	.73	.8	.91	.92	.86	.91	.94	.91	.82	.53	.61	.75
3wloc4-DE-2ALL	w-DE-A	.59	.57	.6	.6	.65	.4	.55	.7	.68	.73	.56	.63	.65	.68	.65
3wloc5-DE-2ALL	All	.67	.75	.78	.75	.82	.62	.74	.78	.71	.76	.72	.8	.79	.82	.83
3wloc5-DE-2ALL	w-DE-C	.94	.92	.89	.87	.91	.95	.79	.81	.72	.74	.88	.96	.97	.96	.96
3wloc5-DE-2ALL	w-DE-E	.73	.84	.93	.91	.96	.45	.8	.84	.75	.81	.76	.93	.88	.91	.96
3wloc5-DE-2ALL	w-DE-A	.35	.51	.51	.48	.59	.45	.63	.69	.66	.72	.51	.5	.52	.61	.56
3wloc6-DE-2ALL	All	.65	.69	.75	.77	.78	.69	.8	.84	.81	.81	.7	.8	.84	.84	.81
3wloc6-DE-2ALL	w-DE-D	.82	.92	.92	.91	.85	.68	.79	.84	.8	.79	.89	.98	.95	.97	.82
3wloc6-DE-2ALL	w-DE-E	.73	.8	.89	.9	.9	.68	.86	.9	.86	.89	.72	.92	.97	.95	.96
3wloc6-DE-2ALL	w-DE-A	.39	.35	.43	.51	.6	.72	.75	.77	.76	.76	.48	.5	.59	.6	.64

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
3wloc7-DE-2ALL	All	.43	.43	.47	.54	.65	.64	.7	.71	.71	.75	.39	.4	.42	.43	.56
3wloc7-DE-2ALL	w-DE-D	.24	.46	.54	.55	.69	.81	.86	.83	.86	.87	.58	.55	.41	.52	.58
3wloc7-DE-2ALL	w-DE-C	.66	.5	.54	.67	.79	.67	.67	.65	.71	.78	.37	.44	.63	.53	.6
3wloc7-DE-2ALL	w-DE-B	.4	.32	.34	.42	.47	.45	.58	.64	.57	.59	.22	.2	.2	.23	.49
3wloc8-DE-2ALL	All	.58	.58	.61	.67	.75	.59	.68	.66	.65	.66	.57	.68	.65	.66	.65
3wloc8-DE-2ALL	w-DE-C	.9	.85	.85	.84	.85	.71	.69	.54	.64	.52	.86	.69	.87	.86	.75
3wloc8-DE-2ALL	w-DE-E	.82	.84	.91	.89	.91	.62	.76	.84	.79	.85	.73	.9	.96	.95	.94
3wloc8-DE-2ALL	w-DE-B	.01	.05	.06	.29	.48	.44	.58	.61	.54	.62	.1	.44	.12	.17	.26
3wloc9-DE-2ALL	All	.55	.59	.62	.64	.68	.71	.81	.83	.8	.84	.58	.64	.63	.68	.74
3wloc9-DE-2ALL	w-DE-D	.82	.89	.77	.48	.79	.6	.81	.83	.76	.82	.85	.92	.67	.86	.7
3wloc9-DE-2ALL	w-DE-E	.8	.84	.89	.91	.91	.86	.86	.94	.87	.89	.75	.94	.92	.96	.97
3wloc9-DE-2ALL	w-DE-B	.02	.06	.2	.53	.35	.68	.75	.71	.75	.82	.14	.06	.31	.2	.55
3wloc10-DE-2ALL	All	.51	.61	.65	.7	.84	.72	.86	.88	.86	.86	.59	.66	.68	.68	.81
3wloc10-DE-2ALL	w-DE-D	.24	.4	.43	.35	.77	.64	.88	.91	.92	.94	.24	.25	.31	.32	.62
3wloc10-DE-2ALL	w-DE-C	.48	.56	.6	.8	.8	.9	.91	.87	.82	.88	.77	.81	.74	.74	.84
3wloc10-DE-2ALL	w-DE-E	.82	.85	.91	.94	.94	.61	.8	.86	.83	.75	.74	.92	.98	.98	.97
Three WiFi locations and any cellular location																
3wlocm1-DE-2ALL	All	.48	.53	.56	.61	.7	.42	.54	.61	.62	.66	.48	.55	.57	.57	.6
3wlocm1-DE-2ALL	mobile	.63	.63	.67	.74	.86	.06	.52	.64	.62	.74	.48	.7	.74	.75	.75
3wlocm1-DE-2ALL	w-DE-C	.85	.69	.78	.73	.78	.63	.67	.66	.66	.67	.89	.95	.88	.77	.6
3wlocm1-DE-2ALL	w-DE-A	.44	.53	.49	.52	.66	.5	.5	.6	.6	.65	.45	.52	.54	.57	.56
3wlocm1-DE-2ALL	w-DE-B	.01	.26	.31	.46	.51	.49	.46	.53	.59	.56	.11	.04	.12	.2	.49
3wlocm1-DE-22	All	.44	.44	.47	.5	.55	.37	.51	.59	.61	.65	.34	.46	.48	.52	.56
3wlocm1-DE-22	mobile	.69	.72	.76	.75	.91	.46	.62	.59	.62	.68	.72	.77	.76	.84	.9
3wlocm1-DE-22	w-DE-C	.46	.57	.43	.37	.37	.13	.53	.61	.57	.74	.4	.21	.37	.38	.47
3wlocm1-DE-22	w-DE-A	.51	.31	.55	.54	.6	.08	.21	.57	.66	.67	.17	.6	.59	.49	.52
3wlocm1-DE-22	w-DE-B	.08	.16	.14	.31	.35	.81	.67	.59	.6	.53	.06	.26	.18	.38	.36
3wlocm2-DE-23	All	.43	.5	.54	.6	.64	.61	.7	.65	.66	.65	.46	.5	.52	.58	.66
3wlocm2-DE-23	mobile	.73	.73	.78	.79	.9	.73	.65	.62	.63	.59	.78	.84	.88	.79	.8
3wlocm2-DE-23	w-DE-D	.07	.12	.17	.36	.44	.72	.75	.74	.77	.81	.34	.39	.39	.53	.63
3wlocm2-DE-23	w-DE-A	.32	.66	.67	.71	.72	.86	.87	.64	.62	.63	.72	.64	.69	.62	.76
3wlocm2-DE-23	w-DE-B	.59	.5	.52	.53	.49	.15	.51	.61	.63	.56	.01	.14	.12	.37	.44

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
3wlocm2-DE-2ALL	All	.49	.55	.55	.61	.67	.54	.64	.69	.71	.73	.46	.53	.56	.59	.68
3wlocm2-DE-2ALL	mobile	.65	.65	.71	.75	.83	.46	.69	.74	.77	.77	.54	.68	.75	.77	.8
3wlocm2-DE-2ALL	w-DE-D	.82	.72	.71	.74	.73	.74	.74	.8	.76	.77	.83	.9	.78	.8	.69
3wlocm2-DE-2ALL	w-DE-A	.45	.51	.5	.53	.61	.43	.45	.54	.6	.66	.41	.49	.51	.56	.59
3wlocm2-DE-2ALL	w-DE-B	.04	.34	.29	.41	.52	.54	.69	.69	.71	.73	.06	.04	.2	.24	.63
3wlocm3-DE-2ALL	All	.45	.59	.61	.67	.77	.49	.65	.71	.71	.73	.52	.6	.65	.69	.71
3wlocm3-DE-2ALL	mobile	.4	.55	.63	.65	.8	.39	.65	.75	.74	.75	.12	.41	.48	.63	.77
3wlocm3-DE-2ALL	w-DE-E	.47	.77	.78	.92	.91	.56	.85	.86	.82	.88	.74	.73	.83	.83	.79
3wlocm3-DE-2ALL	w-DE-A	.32	.51	.52	.52	.59	.39	.49	.54	.59	.61	.4	.49	.5	.48	.54
3wlocm3-DE-2ALL	w-DE-B	.63	.54	.53	.6	.79	.64	.6	.67	.7	.68	.8	.77	.78	.81	.74
3wlocm3-DE-24	All	.46	.57	.63	.77	.77	.56	.74	.79	.79	.77	.47	.57	.63	.69	.7
3wlocm3-DE-24	mobile	.72	.68	.73	.91	.89	.75	.7	.88	.89	.89	.48	.73	.85	.84	.85
3wlocm3-DE-24	w-DE-E	.15	.46	.49	.8	.78	.51	.72	.76	.72	.76	.33	.35	.46	.61	.49
3wlocm3-DE-24	w-DE-A	.57	.77	.77	.79	.78	.66	.81	.81	.82	.8	.71	.76	.77	.88	.87
3wlocm3-DE-24	w-DE-B	.38	.37	.51	.58	.62	.3	.75	.7	.71	.61	.35	.44	.45	.42	.59
3wlocm4-DE-2ALL	All	.48	.53	.58	.6	.77	.54	.67	.72	.73	.76	.49	.55	.58	.6	.7
3wlocm4-DE-2ALL	mobile	.61	.69	.75	.7	.87	.42	.46	.6	.64	.69	.57	.68	.77	.78	.84
3wlocm4-DE-2ALL	w-DE-D	.36	.43	.43	.68	.8	.78	.77	.8	.81	.81	.33	.12	.39	.36	.58
3wlocm4-DE-2ALL	w-DE-C	.56	.46	.62	.46	.79	.51	.89	.77	.84	.86	.61	.87	.62	.7	.79
3wlocm4-DE-2ALL	w-DE-A	.4	.53	.52	.56	.61	.43	.54	.7	.64	.68	.43	.52	.55	.56	.6
3wlocm5-DE-2ALL	All	.53	.62	.68	.71	.79	.48	.67	.72	.67	.69	.56	.65	.74	.71	.75
3wlocm5-DE-2ALL	mobile	.48	.54	.57	.7	.89	.31	.54	.62	.55	.68	.03	.31	.64	.65	.7
3wlocm5-DE-2ALL	w-DE-C	.87	.82	.8	.82	.86	.67	.83	.79	.76	.73	.97	.93	.94	.91	.92
3wlocm5-DE-2ALL	w-DE-E	.36	.7	.81	.87	.86	.47	.74	.82	.78	.68	.72	.82	.84	.82	.84
3wlocm5-DE-2ALL	w-DE-A	.41	.42	.56	.46	.54	.45	.56	.63	.58	.67	.53	.53	.55	.45	.52
3wlocm6-DE-2ALL	All	.54	.64	.65	.72	.73	.57	.71	.78	.73	.77	.53	.65	.7	.68	.7
3wlocm6-DE-2ALL	mobile	.52	.56	.52	.68	.8	.58	.54	.76	.71	.76	.13	.45	.54	.51	.64
3wlocm6-DE-2ALL	w-DE-D	.87	.77	.83	.82	.8	.67	.73	.78	.71	.76	.83	.92	.91	.91	.77
3wlocm6-DE-2ALL	w-DE-E	.4	.74	.77	.89	.8	.54	.81	.87	.83	.8	.75	.73	.81	.87	.79
3wlocm6-DE-2ALL	w-DE-A	.36	.48	.46	.5	.51	.51	.74	.72	.66	.74	.43	.48	.53	.46	.61
3wlocm7-DE-2ALL	All	.39	.45	.48	.55	.68	.51	.62	.65	.65	.67	.39	.44	.44	.5	.57
3wlocm7-DE-2ALL	mobile	.62	.66	.69	.75	.85	.33	.59	.58	.59	.69	.59	.71	.73	.86	.83
3wlocm7-DE-2ALL	w-DE-D	.33	.4	.37	.48	.73	.76	.76	.78	.8	.83	.3	.23	.27	.33	.55
3wlocm7-DE-2ALL	w-DE-C	.52	.51	.57	.63	.78	.58	.78	.67	.66	.58	.66	.78	.62	.62	.57
3wlocm7-DE-2ALL	w-DE-B	.09	.23	.29	.34	.36	.39	.34	.57	.55	.59	.04	.06	.15	.19	.33

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
3wlocm8-DE-2ALL	All	.45	.53	.57	.66	.77	.46	.6	.65	.59	.61	.46	.54	.56	.61	.62
3wlocm8-DE-2ALL	mobile	.38	.47	.51	.68	.85	.2	.58	.64	.64	.66	.14	.37	.53	.68	.71
3wlocm8-DE-2ALL	w-DE-C	.85	.82	.75	.78	.86	.56	.65	.66	.42	.53	.64	.85	.75	.92	.7
3wlocm8-DE-2ALL	w-DE-E	.57	.77	.85	.85	.86	.49	.74	.84	.75	.75	.76	.83	.83	.83	.81
3wlocm8-DE-2ALL	w-DE-B	.01	.06	.18	.33	.51	.59	.42	.46	.55	.52	.29	.11	.12	.03	.27
3wlocm9-DE-2ALL	All	.44	.51	.56	.62	.7	.57	.71	.75	.76	.76	.45	.51	.58	.57	.67
3wlocm9-DE-2ALL	mobile	.32	.5	.52	.74	.82	.43	.63	.65	.72	.63	.19	.4	.62	.53	.68
3wlocm9-DE-2ALL	w-DE-D	.82	.67	.68	.59	.7	.61	.76	.8	.76	.87	.86	.85	.8	.88	.51
3wlocm9-DE-2ALL	w-DE-E	.56	.75	.84	.89	.85	.62	.86	.9	.88	.8	.75	.78	.78	.83	.84
3wlocm9-DE-2ALL	w-DE-B	.04	.13	.19	.24	.44	.6	.6	.64	.66	.72	.01	.03	.13	.03	.64
3wlocm10-DE-2ALL	All	.47	.55	.57	.65	.78	.57	.74	.76	.72	.74	.45	.56	.6	.57	.7
3wlocm10-DE-2ALL	mobile	.43	.54	.54	.62	.77	.21	.57	.59	.58	.61	.15	.38	.52	.55	.64
3wlocm10-DE-2ALL	w-DE-D	.34	.54	.39	.52	.74	.67	.77	.85	.78	.8	.37	.3	.3	.22	.61
3wlocm10-DE-2ALL	w-DE-C	.66	.41	.54	.57	.75	.87	.87	.81	.75	.78	.54	.74	.8	.67	.76
3wlocm10-DE-2ALL	w-DE-E	.44	.74	.81	.89	.88	.51	.76	.79	.79	.76	.75	.8	.78	.85	.81
Four WiFi locations																
4wloc1-DE-2ALL	All	.4	.44	.48	.51	.62	.57	.62	.66	.66	.68	.39	.42	.44	.46	.55
4wloc1-DE-2ALL	w-DE-D	.49	.39	.38	.48	.81	.77	.77	.8	.84	.82	.09	.2	.37	.42	.55
4wloc1-DE-2ALL	w-DE-C	.4	.55	.65	.67	.76	.73	.68	.76	.63	.66	.89	.83	.61	.53	.45
4wloc1-DE-2ALL	w-DE-A	.45	.51	.55	.55	.59	.4	.52	.59	.58	.63	.52	.54	.58	.64	.65
4wloc1-DE-2ALL	w-DE-B	.27	.33	.35	.35	.34	.37	.53	.52	.57	.59	.07	.13	.19	.24	.55
4wloc2-DE-2ALL	All	.51	.55	.59	.6	.69	.51	.59	.63	.58	.62	.53	.6	.62	.62	.64
4wloc2-DE-2ALL	w-DE-C	.82	.91	.78	.79	.86	.67	.64	.65	.58	.49	.82	.93	.83	.93	.8
4wloc2-DE-2ALL	w-DE-E	.73	.85	.87	.88	.88	.48	.77	.82	.74	.85	.76	.9	.94	.89	.94
4wloc2-DE-2ALL	w-DE-A	.41	.41	.54	.46	.53	.39	.52	.54	.57	.57	.51	.51	.54	.55	.56
4wloc2-DE-2ALL	w-DE-B	.1	.03	.15	.27	.5	.5	.43	.52	.44	.59	.06	.04	.17	.09	.29
4wloc3-DE-2ALL	All	.49	.51	.55	.6	.6	.61	.69	.73	.75	.75	.53	.58	.61	.59	.66
4wloc3-DE-2ALL	w-DE-D	.83	.81	.83	.74	.53	.62	.82	.74	.78	.77	.85	.84	.69	.78	.59
4wloc3-DE-2ALL	w-DE-E	.69	.82	.87	.88	.91	.74	.85	.87	.86	.88	.75	.92	.96	.92	.91
4wloc3-DE-2ALL	w-DE-A	.38	.3	.39	.41	.55	.43	.45	.61	.59	.6	.47	.45	.49	.5	.59
4wloc3-DE-2ALL	w-DE-B	.06	.11	.13	.38	.4	.67	.65	.72	.75	.73	.06	.12	.31	.17	.54

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
4wloc4-DE-2ALL	All	.51	.59	.61	.66	.77	.62	.73	.79	.76	.73	.52	.63	.65	.58	.72
4wloc4-DE-2ALL	w-DE-D	.52	.39	.44	.38	.82	.68	.82	.85	.85	.84	.22	.31	.46	.63	.64
4wloc4-DE-2ALL	w-DE-C	.38	.62	.58	.82	.78	.92	.83	.81	.79	.68	.69	.73	.64	.27	.78
4wloc4-DE-2ALL	w-DE-E	.73	.82	.87	.94	.91	.46	.7	.84	.76	.76	.74	.92	.96	.87	.93
4wloc4-DE-2ALL	w-DE-A	.41	.54	.55	.51	.57	.41	.54	.66	.61	.63	.45	.56	.54	.56	.55
4wloc5-DE-2ALL	All	.44	.48	.47	.55	.67	.59	.66	.68	.69	.68	.45	.49	.51	.51	.62
4wloc5-DE-2ALL	w-DE-D	.35	.33	.24	.28	.68	.63	.81	.82	.83	.79	.26	.29	.6	.2	.61
4wloc5-DE-2ALL	w-DE-C	.62	.68	.72	.83	.79	.76	.68	.58	.65	.59	.71	.64	.27	.73	.52
4wloc5-DE-2ALL	w-DE-E	.79	.87	.89	.88	.88	.57	.73	.82	.78	.78	.76	.94	.96	.96	.99
4wloc5-DE-2ALL	w-DE-B	.0	.03	.04	.21	.31	.39	.44	.52	.49	.55	.08	.08	.22	.13	.37
Four WiFi locations and any cellular location																
4wlocm1-DE-2ALL	All	.38	.43	.47	.53	.64	.47	.57	.64	.63	.65	.39	.45	.49	.48	.57
4wlocm1-DE-2ALL	mobile	.62	.63	.64	.72	.83	.16	.48	.61	.59	.66	.58	.71	.78	.71	.8
4wlocm1-DE-2ALL	w-DE-D	.42	.33	.47	.39	.74	.75	.76	.76	.83	.81	.14	.12	.5	.32	.55
4wlocm1-DE-2ALL	w-DE-C	.45	.55	.52	.68	.71	.5	.7	.72	.62	.71	.82	.87	.53	.64	.66
4wlocm1-DE-2ALL	w-DE-A	.42	.48	.5	.56	.53	.47	.48	.57	.59	.59	.42	.52	.54	.58	.56
4wlocm1-DE-2ALL	w-DE-B	.0	.17	.22	.31	.37	.46	.44	.55	.52	.48	.0	.02	.1	.13	.29
4wlocm2-DE-2ALL	All	.43	.5	.53	.62	.67	.42	.57	.63	.6	.6	.41	.51	.55	.57	.57
4wlocm2-DE-2ALL	mobile	.39	.47	.51	.78	.8	.28	.56	.67	.63	.66	.09	.37	.52	.66	.62
4wlocm2-DE-2ALL	w-DE-C	.95	.74	.74	.73	.81	.45	.63	.63	.56	.61	.86	.86	.86	.69	.59
4wlocm2-DE-2ALL	w-DE-E	.41	.69	.77	.84	.85	.47	.72	.8	.76	.63	.74	.78	.78	.8	.81
4wlocm2-DE-2ALL	w-DE-A	.41	.51	.54	.48	.47	.44	.53	.55	.59	.68	.3	.5	.51	.52	.46
4wlocm2-DE-2ALL	w-DE-B	.01	.11	.08	.25	.42	.47	.43	.52	.46	.43	.03	.03	.06	.19	.36
4wlocm3-DE-2ALL	All	.4	.52	.53	.57	.64	.5	.65	.68	.68	.69	.43	.53	.56	.56	.65
4wlocm3-DE-2ALL	mobile	.44	.61	.65	.6	.78	.34	.52	.66	.65	.61	.13	.58	.47	.56	.64
4wlocm3-DE-2ALL	w-DE-D	.73	.68	.68	.76	.59	.66	.73	.73	.68	.74	.81	.88	.74	.74	.66
4wlocm3-DE-2ALL	w-DE-E	.43	.77	.75	.88	.86	.47	.82	.84	.81	.85	.76	.68	.82	.84	.83
4wlocm3-DE-2ALL	w-DE-A	.35	.46	.44	.47	.53	.46	.49	.55	.58	.59	.39	.48	.59	.54	.55
4wlocm3-DE-2ALL	w-DE-B	.06	.07	.13	.13	.44	.6	.66	.62	.69	.69	.03	.01	.2	.13	.59
4wlocm4-DE-2ALL	All	.4	.53	.56	.62	.75	.49	.66	.72	.69	.69	.43	.52	.57	.58	.63
4wlocm4-DE-2ALL	mobile	.29	.58	.55	.64	.86	.01	.41	.56	.56	.69	.15	.29	.57	.58	.59
4wlocm4-DE-2ALL	w-DE-D	.3	.32	.17	.44	.8	.68	.78	.79	.74	.8	.27	.26	.33	.55	.59
4wlocm4-DE-2ALL	w-DE-C	.52	.63	.73	.69	.8	.9	.85	.88	.79	.67	.59	.69	.71	.39	.66
4wlocm4-DE-2ALL	w-DE-E	.5	.68	.8	.87	.82	.46	.71	.77	.75	.73	.73	.86	.81	.83	.79
4wlocm4-DE-2ALL	w-DE-A	.36	.44	.56	.46	.48	.42	.56	.59	.62	.53	.4	.51	.45	.54	.54

Continued on next page

Table C.3 – continued from previous page

Classification	Rec.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
4wlocm5-DE-2ALL	All	.38	.44	.47	.54	.66	.47	.63	.64	.65	.68	.37	.42	.45	.5	.54
4wlocm5-DE-2ALL	mobile	.58	.56	.58	.66	.81	.17	.57	.59	.58	.67	.23	.34	.53	.65	.55
4wlocm5-DE-2ALL	w-DE-D	.43	.46	.49	.31	.59	.64	.78	.8	.78	.8	.19	.14	.19	.36	.52
4wlocm5-DE-2ALL	w-DE-C	.47	.39	.37	.67	.79	.71	.57	.58	.64	.66	.6	.84	.66	.63	.7
4wlocm5-DE-2ALL	w-DE-E	.41	.79	.8	.92	.82	.46	.72	.83	.8	.8	.75	.79	.82	.8	.82
4wlocm5-DE-2ALL	w-DE-B	.0	.02	.09	.16	.26	.38	.51	.4	.45	.5	.08	.01	.05	.04	.12
Five WiFi locations																
5wloc-DE-2ALL	All	.41	.46	.48	.54	.65	.52	.63	.66	.61	.6	.41	.46	.49	.51	.56
5wloc-DE-2ALL	w-DE-D	.62	.19	.33	.26	.63	.7	.76	.78	.77	.78	.14	.28	.3	.58	.59
5wloc-DE-2ALL	w-DE-C	.36	.65	.61	.79	.83	.64	.66	.61	.54	.5	.71	.6	.64	.49	.49
5wloc-DE-2ALL	w-DE-E	.71	.81	.83	.94	.87	.42	.75	.83	.75	.74	.74	.91	.93	.92	.94
5wloc-DE-2ALL	w-DE-A	.36	.52	.41	.44	.56	.43	.49	.55	.55	.44	.46	.43	.51	.58	.51
5wloc-DE-2ALL	w-DE-B	.0	.13	.2	.26	.36	.43	.5	.51	.46	.54	.01	.09	.07	.01	.26
Five WiFi locations and any cellular location																
5wlocm-DE-2ALL	All	.35	.44	.43	.53	.63	.44	.57	.62	.58	.62	.37	.43	.46	.49	.56
5wlocm-DE-2ALL	mobile	.35	.5	.54	.66	.76	.18	.47	.57	.49	.69	.19	.36	.59	.65	.71
5wlocm-DE-2ALL	w-DE-D	.2	.25	.21	.37	.59	.66	.7	.76	.68	.73	.39	.17	.22	.56	.41
5wlocm-DE-2ALL	w-DE-C	.69	.75	.5	.63	.76	.52	.67	.5	.54	.66	.35	.73	.67	.46	.51
5wlocm-DE-2ALL	w-DE-E	.48	.72	.77	.87	.85	.45	.73	.8	.73	.65	.74	.79	.79	.72	.79
5wlocm-DE-2ALL	w-DE-A	.37	.42	.41	.44	.46	.4	.52	.58	.58	.54	.36	.51	.48	.54	.53
5wlocm-DE-2ALL	w-DE-B	.01	.01	.16	.21	.36	.43	.36	.53	.47	.44	.19	.0	.01	.02	.43
All possible locations (WiFi and cellular)																
all-DE-2ALL	All	.29	.38	.42	.45	.6	.39	.55	.56	.56	.54	.36	.46	.48	.49	.53
all-DE-2ALL	m-DE-A	.5	.35	.43	.45	.7	.42	.47	.43	.48	.32	.58	.6	.66	.73	.81
all-DE-2ALL	m-DE-B	.05	.39	.3	.42	.53	.31	.55	.68	.69	.59	.0	.24	.47	.51	.48
all-DE-2ALL	w-DE-D	.21	.5	.55	.24	.61	.63	.73	.68	.74	.7	.07	.35	.53	.49	.48
all-DE-2ALL	w-DE-C	.47	.41	.37	.61	.83	.29	.72	.28	.28	.21	.76	.63	.36	.43	.42
all-DE-2ALL	w-DE-E	.43	.64	.74	.82	.78	.47	.67	.81	.79	.72	.73	.74	.76	.73	.76
all-DE-2ALL	w-DE-A	.37	.39	.46	.44	.52	.09	.36	.57	.48	.66	.31	.61	.49	.51	.51
all-DE-2ALL	w-DE-B	.03	.0	.07	.18	.26	.51	.37	.49	.44	.56	.08	.02	.08	.01	.26
Network connection (WiFi vs. cellular) of receiver devices																
network-DE-2ALL	All	.72	.77	.79	.85	.91	.63	.75	.81	.82	.85	.68	.8	.83	.85	.88
network-DE-2ALL	mobile	.72	.7	.75	.83	.92	.58	.79	.8	.82	.84	.57	.74	.82	.83	.89
network-DE-2ALL	wifi	.72	.83	.84	.86	.9	.68	.72	.82	.82	.87	.79	.85	.84	.88	.86

Continued on next page

Table C.3 – continued from previous page

Classification	**Rec.**	**Signal**					**Threema**					**Whatsapp**				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
network-DE-22	All	.78	.8	.81	.82	.92	.74	.81	.86	.88	.9	.69	.83	.87	.91	.94
network-DE-22	mobile	.73	.7	.75	.82	.94	.72	.81	.87	.87	.89	.8	.78	.83	.88	.93
network-DE-22	wifi	.82	.9	.87	.82	.9	.76	.8	.85	.89	.91	.59	.89	.92	.94	.96
network-DE-23	All	.79	.79	.82	.84	.9	.75	.82	.77	.77	.75	.82	.83	.86	.86	.9
network-DE-23	mobile	.72	.75	.8	.8	.9	.77	.85	.87	.91	.74	.84	.89	.91	.92	.92
network-DE-23	wifi	.87	.84	.84	.88	.9	.74	.8	.68	.63	.76	.8	.78	.8	.81	.87
network-DE-24	All	.76	.85	.87	.93	.95	.8	.87	.91	.92	.94	.79	.87	.88	.91	.92
network-DE-24	mobile	.91	.86	.87	.93	.95	.92	.9	.91	.93	.94	.98	.92	.88	.92	.93
network-DE-24	wifi	.61	.85	.88	.93	.95	.69	.84	.91	.91	.94	.6	.83	.89	.9	.91

Bibliography

[1] K. Abe and S. Goto, "Fingerprinting Attack on Tor Anonymity using Deep Learning," in *Asia-Pacific Advanced Network Research Workshop*, ser. APAN '16. APAN, 2016, pp. 15–20.

[2] A. Abouzied and J. Chen, "Harnessing Data Loss With Forgetful Data Structures," in *ACM Symposium on Cloud Computing*, ser. SoCC '15. Kohala Coast, HI, USA: ACM, Aug. 2015, pp. 168–173.

[3] A. Adams and M. A. Sasse, "Users Are Not The Enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, Dec. 1999.

[4] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, "Over-exposed? Privacy Patterns and Considerations in Online and Mobile Photo Sharing," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '07. San Jose, CA, USA: ACM, Apr. 2007, pp. 357–366.

[5] M. Akhoondi, C. Yu, and H. V. Madhyastha, "LASTor: A Low-Latency AS-Aware Tor Client," in *IEEE Symposium on Security and Privacy*, ser. SP '12. San Francisco, CA, USA: IEEE, May 2012, pp. 476–490.

[6] H. Almuhimedi, S. Wilson, B. Liu, N. Sadeh, and A. Acquisti, "Tweets are Forever: A Large-scale Quantitative Analysis of Deleted Tweets," in *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, ser. CSCW '13. San Antonio, TX, USA: ACM, Feb. 2013, pp. 897–908.

[7] A. Alqhatani and H. R. Lipford, ""There is nothing that I need to keep secret": Sharing Practices and Concerns of Wearable Fitness Data," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '19. Santa Clara, CA, USA: USENIX Association, Aug. 2019.

[8] G. Amjad, M. S. Mirza, and C. Pöpper, "Forgetting with Puzzles: Using Cryptographic Puzzles to support Digital Forgetting," in *ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '18. Tempe, AZ, USA: ACM, Mar. 2018, pp. 342–353.

[9] R. Ariano, "'What do the check marks mean on WhatsApp?': How to determine the status of your message on WhatsApp," Jan. 2020, https://www.businessinsider.com/what-do-the-check-marks-mean-on-whatsapp, as of December 2, 2022.

[10] O. Ayalon and E. Toch, "Retrospective Privacy: Managing Longitudinal Privacy in Online Social Networks," in *ACM Symposium On Usable Privacy and Security*, ser. SOUPS '13. Newcastle, UK: ACM, Jul. 2013.

[11] ——, "Not Even Past: Information Aging and Temporal Privacy in Online Social Networks," *Human-Computer Interaction*, vol. 32, no. 2, pp. 73–102, Oct. 2017.

[12] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, "Mix&Slice: Efficient Access Revocation in the Cloud," in *ACM Conference on Computer and Communications Security*, ser. CCS '16. Vienna, Austria: ACM, Oct. 2016, pp. 217–228.

[13] A. Back, U. Möller, and A. Stiglic, "Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems," in *International Workshop on Information Hiding*, ser. IWIH '01. Pittsburgh, PA, USA: Springer, Apr. 2001, pp. 245–257.

[14] A. Bahramali, A. Houmansadr, R. Soltani, D. Goeckel, and D. Towsley, "Practical Traffic Analysis Attacks on Secure Messaging Applications," in *Network and Distributed System Security Symposium*, ser. NDSS '20. San Diego, CA, USA: The Internet Society, Feb. 2020.

[15] S. Barth and M. D. de Jong, "The Privacy Paradox: "Investigating Discrepancies Between Expressed Privacy Concerns and Actual Online Behavior" A Systematic Literature Review," *Telematics and Informatics*, vol. 34, no. 7, pp. 1038–1058, Nov. 2017.

[16] A. Barton and M. Wright, "DeNASA: Destination-Naive AS-Awareness in Anonymous Communications," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '16. Darmstadt, Germany: De Gruyter, Oct. 2016, pp. 356–372.

[17] L. Bauer, L. F. Cranor, S. Komanduri, M. L. Mazurek, M. K. Reiter, M. Sleeper, and B. Ur, "The Post Anachronism: The Temporal Dimension of Facebook Privacy," in *Workshop on Privacy in the Electronic Society*, ser. WPES '13. Berlin, Germany: ACM, Nov. 2013, pp. 1–12.

[18] J. B. Bayer, N. B. Ellison, S. Y. Schoenebeck, and E. B. Falk, "Sharing the Small Moments: Ephemeral Social Interaction on Snapchat," *Information, Communication & Society*, vol. 19, no. 7, pp. 956–977, Apr. 2016.

[19] F. Beato, M. Kohlweiss, and K. Wouters, "Scramble! Your Social Network Data," in *Privacy Enhancing Technologies Symposium*, ser. PETS '11, Waterloo, Canada, Jul. 2011, pp. 211–225.

[20] D. J. Bernstein, M. Hamburg, A. Krasnova, and T. Lange, "Elligator: Elliptic-Curve Points Indistinguishable from Uniform Random Strings," in *ACM Conference on Computer and Communications Security*, ser. CCS '13. Berlin, Germany: ACM, Nov. 2013, pp. 967–980.

[21] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer, "Quantifying the Invisible Audience in Social Networks," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '13. Paris, France: ACM, Apr. 2013, pp. 21—-30.

[22] O. Berthold and H. Langos, "Dummy Traffic Against Long Term Intersection Attacks," in *Privacy Enhancing Technologies Workshop*, ser. PET '02. San Francisco, CA, USA: Springer, Apr. 2002, pp. 110–128.

[23] A. Besmer and H. Richter Lipford, "Moving Beyond Untagging: Photo Privacy in a Tagged World," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '10. Atlanta, GA, USA: ACM, Apr. 2010, pp. 1563–1572.

[24] P. Bhattacharya and N. Ganguly, "Characterizing Deleted Tweets and Their Authors," in *AAAI Conference on Weblogs and Social Media*, ser. ICWSM '16. Cologne, Germany: AAAI, May 2016.

[25] M. Bishop, E. R. Butler, K. Butler, C. Gates, and S. Greenspan, "Forgive and Forget: Return to Obscurity," in *New Security Paradigms Workshop*, ser. NSPW '13. Banff, Canada: ACM, Sep. 2013, pp. 1–10.

[26] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy Vulnerabilities in Encrypted HTTP Streams," in *Privacy Enhancing Technologies Workshop*, ser. PET '05. Cavtat, Croatia: Springer, May 2005, pp. 1–11.

[27] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting Inside Attackers Using Decoy Documents," in *International Conference on Security and Privacy in Communication Systems*, ser. SecureComm '09, Athens, Greece, Sep. 2009, pp. 51–70.

[28] T. F. Brady, T. Konkle, and G. A. Alvarez, "A Review of Visual Memory Capacity: Beyond Individual Items and Toward Structured Representations," *Journal of Vision*, vol. 11, no. 5, pp. 1–34, May 2011.

[29] V. Buterin, "A Next-generation Smart Contract and Decentralized Application Platform," 2014, https://github.com/ethereum/wiki/wiki/White-Paper, as of December 2, 2022.

[30] X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense," in *Workshop on Privacy in the Electronic Society*, ser. WPES '14. Scottsdale, AZ, USA: ACM, Nov. 2014, pp. 121–130.

[31] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses," in *ACM Conference on Computer and Communications Security*, ser. CCS '14. Scottsdale, AZ, USA: ACM, Nov. 2014, pp. 227–238.

[32] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a Distance: Website Fingerprinting Attacks and Defenses," in *ACM Conference on Computer and Communications Security*, ser. CCS '12. Raleigh, NC, USA: ACM, Oct. 2012, pp. 605–616.

[33] Y. Cao and J. Yang, "Towards Making Systems Forget With Machine Unlearning," in *IEEE Symposium on Security and Privacy*, ser. S&P '15, San Jose, CA, USA, May 2015, pp. 463–480.

[34] B. Carminati and E. Ferrari, "Collaborative Access Control in Online Social Networks," in *Conference on Collaborative Computing: Networking, Applications and Worksharing*, ser. CollaborateCom '11. Orlando, FL, USA: IEEE, Oct. 2011, pp. 231–240.

[35] C. Castelluccia, E. De Cristofaro, A. Francillon, and M.-A. Kaafar, "EphPub: Toward Robust Ephemeral Publishing," in *IEEE Conference on Network Protocols*, ser. ICNP '11. Vancouver, BC, Canada: IEEE, Oct. 2011, pp. 165–175.

[36] I. Cervesato, "The Dolev-Yao Intruder is the Most Powerful Attacker," in *Annual Symposium on Logic in Computer Science*, ser. LICS '01. Boston, MA, USA: IEEE, Jun. 2001.

[37] Check-host.net, "API Overview," https://check-host.net/about/api, as of December 2, 2022.

[38] H. Cheng and R. Avnur, "Traffic Analysis of SSL Encrypted Web Browsing," 1998.

[39] G. Cherubin, J. Hayes, and M. Juarez, "Website Fingerprinting Defenses at the Application Layer," in *Privacy Enhancing Technologies Symposium*, ser. PETS '17. Minneapolis, MN, USA: De Gruyter, Jul. 2017, pp. 186–203.

[40] J. W. Clark, P. Snyder, D. McCoy, and C. Kanich, ""I Saw Images I Didn't Even Know I Had": Understanding User Perceptions of Cloud Storage Privacy," in

ACM CHI Conference on Human Factors in Computing Systems, ser. CHI '15. Seoul, Republic of Korea: ACM, Apr. 2015, pp. 1641–1644.

[41] D. Cohen, "Instagram Just Revamped Instagram Direct, Which Now Has 375 Million Monthly Users," Apr. 2017, https://www.adweek.com/digital/instagram-direct-disappearing-photos-videos-375-million-monthly-users/, as of December 2, 2022.

[42] K. Cohn-Gordon, C. J. F. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol," in *IEEE European Symposium on Security and Privacy*, ser. EuroS&P '17. Paris, France: IEEE, Apr. 2017, pp. 451–466.

[43] C. O. Community, "ETH Gas Station," 2017, https://ethgasstation.info/, as of December 2, 2022.

[44] K. P. Coopamootoo and T. Groß, "Why Privacy is All But Forgotten: An Empirical Study of Privacy & Sharing Attitude," in *Privacy Enhancing Technologies Symposium*, ser. PETS '17. Minneapolis, MN, USA: Sciendo, Jul. 2017, pp. 97–118.

[45] S. E. Coull and K. P. Dyer, "Traffic Analysis of Encrypted Messaging Services: Apple iMessage and Beyond," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 6–11, Oct. 2014.

[46] Court of Justice of the European Union, "An internet search engine operator is responsible for the processing that it carries out of personal data which appear on web pages published by third parties," May 2014, https://curia.europa.eu/jcms/upload/docs/application/pdf/2014-05/cp140070en.pdf, as of December 2, 2022.

[47] N. Dalmeijer and V. Niculescu-Dinca, "What's up with WhatsApp Neighbourhood Watch?" Feb. 2019, https://www.leidensecurityandglobalaffairs.nl/articles/whats-up-with-whatsapp-neighbourhood-watch, as of December 2, 2022.

[48] G. Danezis, "The traffic analysis of continuous-time mixes," in *Privacy Enhancing Technologies Workshop*, ser. PET '04. Toronto, Canada: Springer, May 2004, pp. 35–50.

[49] G. Danezis, C. Diaz, and C. Troncoso, "Two-Sided Statistical Disclosure Attack," in *Privacy Enhancing Technologies Symposium*, ser. PETS '07. Ottawa, Canada: Springer, Jun. 2007, pp. 30–44.

[50] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: Privacy at the time of twitter," in *IEEE Symposium on Security and Privacy*, ser. S&P '12. San Francisco, CA, USA: IEEE, May 2012, pp. 285–299.

[51] A. Dhir, P. Kaur, K. Lonka, and M. Nieminen, "Why Do Adolescents Untag Photos on Facebook?" *Computers in Human Behavior*, vol. 55, pp. 1106–1115, Feb. 2016.

[52] R. Dingledine and N. Mathewson, "Tor Path Specification," Dec. 2002, https://gitweb.torproject.org/torspec.git/tree/path-spec.txt, as of December 2, 2022.

[53] C. Duckett and S. J. Vaughan-Nichols, "AWS EC2 North Virginia outage resolves but some issues linger," Sep. 2021, https://www.zdnet.com/article/aws-ec2-north-virginia-outage-resolves-but-some-issues-linger/, as of December 2, 2022.

[54] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail," in *IEEE Symposium on Security and Privacy*, ser. SP '12. San Francisco, CA, USA: IEEE, May 2012, pp. 332–346.

[55] M. Edman and P. Syverson, "AS-Awareness in Tor Path Selection," in *ACM Conference on Computer and Communications Security*, ser. CCS '09. Chicago, IL, USA: ACM, Nov. 2009, pp. 380–389.

[56] S. Egelman, A. Oates, and S. Krishnamurthi, "Oops, I Did it Again: Mitigating Repeated Access Control Errors on Facebook," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '11. Vancouver, Canada: ACM, May 2011, pp. 2295–2304.

[57] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor," in *Workshop on Privacy in the Electronic Society*, ser. WPES '12. Raleigh, NC, USA: ACM, Oct. 2012.

[58] S. Ellis, A. Juels, and S. Nazarov, "ChainLink - A Decentralized Oracle Network," Sep. 2017, https://crushcrypto.com/wp-content/uploads/2017/09/LINK-Whitepaper.pdf, as of December 2, 2022.

[59] A. Etzioni and R. Bhat, "Second Chances, Social Forgiveness, and the Internet," Mar. 2009, https://theamericanscholar.org/second-chances-social-forgiveness-and-the-internet/, as of December 2, 2022.

[60] European Parliament, "Regulation (EU) 2016/679 of the European Parliament and of the Council," May 2016, https://eur-lex.europa.eu/eli/reg/2016/679/oj, as of December 2, 2022.

[61] Facebook, Inc., "Data Centers," 2020, https://sustainability.fb.com/data-centers/, as of December 2, 2022.

[62] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic," in *ACM Internet Measurement Conference*, ser. IMC '20. Virtual Event: ACM, Oct. 2020, pp. 1–18.

[63] C. Fiesler, M. Dye, J. L. Feuston, C. Hiruncharoenvate, C. J. Hutto, S. Morrison, P. Khanipour Roshan, U. Pavalanathan, A. S. Bruckman, M. De Choudhury, and E. Gilbert, "What (or Who) is Public? Privacy Settings and Social Media Content Sharing," in *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, ser. CSCW '17. Portland, OR, USA: ACM, Feb. 2017, pp. 567–580.

[64] T. Frosch, C. Mainka, C. Bader, F. Bergsma, J. Schwenk, and T. Holz, "How Secure is TextSecure?" in *IEEE European Symposium on Security and Privacy*, ser. EuroSP '16. Saarbrücken, Germany: IEEE, Mar. 2016, pp. 457–472.

[65] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data," in *USENIX Security Symposium*, ser. SSYM '09. Montreal, Canada: USENIX Association, Aug. 2009, pp. 299–316.

[66] R. Geambasu, T. Kohno, A. Krishnamurthy, A. Levy, H. M. Levy, P. Gardner, and V. Moscaritolo, "New Directions for Self-Destructing Data," University of Washington, Tech. Rep., 2011.

[67] J. Geddes, R. Jansen, and N. Hopper, "How Low Can You Go: Balancing Performance with Anonymity in Tor," in *Privacy Enhancing Technologies Symposium*, ser. PETS '13. Bloomington, IN, USA: Springer, Jul. 2013, pp. 164–184.

[68] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making AI Forget You: Data Deletion in Machine Learning," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '19. Vancouver, Canada: Curran Associates, Inc., Dec. 2019, pp. 3513–3526.

[69] B. Glanz, T. Sickert, and M. Richter, "Was "Das Netz vergisst nie" für die junge Generation bedeutet"," Oct. 2021, https://www.deutschlandfunkkultur.de/

der-fall-sarah-lee-heinrich-was-das-netz-vergisst-nie-fuer-100.html/, as of December 2, 2022.

[70] Google LLC, "Requests to delist content under European privacy law," https://transparencyreport.google.com/eu-privacy/overview, as of December 2, 2022.

[71] B. Greschbach, G. Kreitz, and S. Buchegger, "The Devil is in the Metadata — New Privacy Challenges in Decentralised Online Social Networks," in *Conference on Pervasive Computing and Communications Workshops*, ser. PerCOM '12. Lugano, Switzerland: IEEE, Mar. 2012, pp. 333–339.

[72] Y. Gurevich, E. Hudis, and J. M. Wing, "Inverse Privacy," *Communications of the ACM*, vol. 59, no. 7, pp. 38–42, Jun. 2016.

[73] H. Habib, N. Shah, and R. Vaish, "Impact of Contextual Factors on Snapchat Public Sharing," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. Glasgow, UK: ACM, May 2019.

[74] R. Hasan, E. Hassan, Y. Li, K. Caine, D. J. Crandall, R. Hoyle, and A. Kapadia, "Viewer Experience of Obscuring Scene Elements in Photos to Enhance Privacy," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. Montreal, Canada: ACM, Apr. 2018.

[75] R. Hasan, Y. Li, E. Hassan, K. Caine, D. J. Crandall, R. Hoyle, and A. Kapadia, "Can Privacy Be Satisfying? On Improving Viewer Satisfaction for Privacy-Enhanced Photos Using Aesthetic Transforms," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. Glasgow, UK: ACM, May 2019.

[76] J. Hayes and G. Danezis, "Guard Sets for Onion Routing," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '15. Philadelphia, PA, USA: De Gruyter, Jul. 2015, pp. 65–80.

[77] ——, "k-Fingerprinting: A Robust Scalable Website Fingerprinting Technique," in *USENIX Security Symposium*, ser. USENIX '16. Austin, TX, USA: USENIX Association, Aug. 2016, pp. 1187–1203.

[78] D. Herrmann, R. Wendolsky, and H. Federrath, ""Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier"," in *ACM Workshop on Cloud Computing Security*, ser. CCSW '09. Chicago, IL, USA: ACM, Nov. 2009, pp. 31–42.

[79] A. Hintz, "Fingerprinting Websites Using Traffic Analysis," in *Privacy Enhancing Technologies Workshop*, ser. PET '02. San Francisco, CA, USA: Springer, Apr. 2002, pp. 171–178.

[80] A. Houmansadr and N. Borisov, "SWIRL: A Scalable Watermark to Detect Correlated Network Flows," in *Network and Distributed System Security Symposium*, ser. NDSS '11. San Diego, CA, USA: The Internet Society, Feb. 2011.

[81] ——, "The need for Flow Fingerprints to Link Correlated Network Flows," in *Privacy Enhancing Technologies Symposium*, ser. PETS '13. Bloomington, IN, USA: Springer, Jul. 2013, pp. 205–224.

[82] A. Houmansadr, N. Kiyavash, and N. Borisov, "RAINBOW: A Robust And Invisible Non-Blind Watermark for Network Flows," in *Network and Distributed System Security Symposium*, ser. NDSS '09. San Diego, CA, USA: The Internet Society, Feb. 2009.

[83] ——, "Non-Blind Watermarking of Network Flows," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1232–1244, Aug. 2014.

[84] E. Hunt, "I Sent A Compromising Message to the Wrong Person. How Will I Ever Recover?" Apr. 2017, https://www.theguardian.com/culture/2017/apr/28/i-sent-a-compromising-message-to-the-wrong-person-how-will-i-ever-recover, as of December 2, 2022.

[85] P. Ilia, B. Carminati, E. Ferrari, P. Fragopoulou, and S. Ioannidis, "SAMPAC: Socially-Aware Collaborative Multi-Party Access Control," in *ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '17. Scottsdale, AZ, USA: ACM, Mar. 2017, pp. 71–82.

[86] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann, "The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network," in *Network and Distributed System Security Symposium*, ser. NDSS '14. San Diego, CA, USA: The Internet Society, Feb. 2014.

[87] R. Jansen, T. Vaidya, and M. Sherr, "Point Break: A Study of Bandwidth Denial-of-Service Attacks against Tor," in *USENIX Security Symposium*, ser. USENIX '19. Santa Clara, CA, USA: USENIX Association, Aug. 2019, pp. 1823–1840.

[88] C. Johansen, A. Mujaj, H. Arshad, and J. Noll, "Comparing Implementations of Secure Messaging Protocols (long version)," University of Oslo, Tech. Rep., 2017.

[89] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, "Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries," in *ACM Conference on Computer and Communications Security*, ser. CCS '13. Berlin, Germany: ACM, Nov. 2013, pp. 337–348.

[90] M. Johnson, S. Egelman, and S. M. Bellovin, "Facebook and Privacy: It's Complicated," in *ACM Symposium On Usable Privacy and Security*, ser. SOUPS '12. Washington, D. C., USA: ACM, Jul. 2012.

[91] A. Johnston, "Now That's Awkward! Hilarious Texts Show What Happens When You Send a Message to the WRONG Person," Oct. 2016, http://www.dailymail.co.uk/femail/article-3871302/Hilarious-texts-happens-send-message-WRONG-person.html, as of December 2, 2022.

[92] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A Critical Evaluation of Website Fingerprinting Attacks," in *ACM Conference on Computer and Communications Security*, ser. CCS '14. Scottsdale, AZ, USA: ACM, Nov. 2014, pp. 263–274.

[93] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an Efficient Website Fingerprinting Defense," in *European Symposium on Research in Computer Security*, ser. ESORICS '16. Heraklion, Greece: Springer, Sep. 2016, pp. 27–46.

[94] J. Juen, A. Johnson, A. Das, N. Borisov, and M. Caesar, "Defending Tor from Network Adversaries: A Case Study of Network Path Prediction," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '15. Philadelphia, PA, USA: De Gruyter, Jun. 2015, pp. 171–187.

[95] R. Kang, L. Dabbish, N. Fruchter, and S. Kiesler, ""My Data Just Goes Everywhere:" User Mental Models of the Internet and Implications for Privacy and Security," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '15. Ottawa, ON, Canada: USENIX Association, Jul. 2015, pp. 39–52.

[96] Kepios Pte. Ltd., "Digital 2021 October Global Statshot," Oct. 2021, https://datareportal.com/reports/digital-2021-october-global-statshot/, as of December 2, 2022.

[97] M. T. Khan, M. Hyun, C. Kanich, and B. Ur, "Forgotten But Not Gone: Identifying the Need for Longitudinal Data Management in Cloud Storage," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. Montreal, Canada: ACM, Apr. 2018.

[98] K. Kohls, K. Jansen, D. Rupprecht, T. Holz, and C. Pöpper, "On the Challenges of Geographical Avoidance for Tor," in *Network and Distributed System Security Symposium*, ser. NDSS '19. San Diego, CA, USA: The Internet Society, Feb. 2019.

[99] K. Kohls and C. Pöpper, "DigesTor: Comparing Passive Traffic Analysis Attacks on Tor," in *European Symposium on Research in Computer Security*, ser. ESORICS '18. Barcelona, Spain: Springer, Sep. 2018, pp. 512–530.

[100] K. Krol, J. M. Spring, S. Parkin, and M. A. Sasse, "Towards robust experimental design for user studies in security and privacy," in *Learning from Authoritative Security Experiment Results*, ser. LASER '16. San Jose, CA, USA: USENIX Association, May 2016, pp. 21–31.

[101] B. Lampson, "Usable Security: How to Get It," *Communications of the ACM*, vol. 52, no. 1, pp. 25–27, Nov. 2009.

[102] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. Los Angeles, CA, USA: ACM, Aug. 2017, pp. 183–196.

[103] H. Lee, D. Kim, and Y. Kwon, "TLS 1.3 in Practice: How TLS 1.3 Contributes to the Internet," in *The Web Conference*, ser. WWW '21. Ljubljana, Slovenia: ACM, Apr. 2021, pp. 70–79.

[104] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing Attacks in Low-Latency Mix Systems," in *International Conference on Financial Cryptography and Data Security*, ser. FC '04. Key West, FL, USA: Springer, Feb. 2004, pp. 251–265.

[105] Y. Li, N. Vishwamitra, B. P. Knijnenburg, H. Hu, and K. Caine, "Blur vs. Block: Investigating the Effectiveness of Privacy-Enhancing Obfuscation for Images," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '17. Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1343–1351.

[106] ——, "Effectiveness and Users' Experience of Obfuscation as a Privacy-Enhancing Technology for Sharing Photos," in *ACM Conference on Computer-*

Supported Cooperative Work and Social Computing, ser. CSCW '18. Jersey City, NJ, USA: ACM, Nov. 2018, pp. 1–24.

[107] M. Liberatore and B. N. Levine, "Inferring the Source of Encrypted HTTP Connections," in *ACM Conference on Computer and Communications Security*, ser. CCS '06. Alexandria, VA, USA: ACM, Oct. 2006, pp. 255–263.

[108] Z. Ling, X. Fu, W. Jia, W. Yu, D. Xuan, and J. Luo, "Novel Packet Size-Based Covert Channel Attacks Against Anonymizer," in *IEEE International Conference on Computer Communications*, ser. INFOCOM '13. Shanghai, China: IEEE, Apr. 2013, pp. 2411–2426.

[109] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A New Cell Counter Based Attack Against Tor," in *ACM Conference on Computer and Communications Security*, ser. CCS '09. Chicaco, IL, USA: ACM, Nov. 2009, pp. 578–589.

[110] H. R. Lipford, A. Besmer, and J. Watson, "Understanding Privacy Settings in Facebook with an Audience View," in *USENIX Workshop on Usability, Psychology, and Security*, ser. UPSEC '08. San Francisco, CA, USA: USENIX Association, Apr. 2008, pp. 1–8.

[111] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Analyzing Facebook Privacy Settings: User Expectations vs. Reality," in *ACM Internet Measurement Conference*, ser. IMC '11. Berlin, Germany: ACM, Nov. 2011, pp. 61–70.

[112] Y. Liu, C. Kliman-Silver, and A. Mislove, "The Tweets They Are a-Changin': Evolution of Twitter Users and Behavior," in *AAAI Conference on Weblogs and Social Media*, ser. ICWSM '14. Ann Arbor, MI, USA: AAAI, Jun. 2014.

[113] F. Lobo, "How WhatsApp brings my family closer together," Apr. 2017, https://mashable.com/2017/04/18/whatsapp-families-india/, as of December 2, 2022.

[114] P. Loch, "WhatsApp in the workplace," Jun. 2019, https://www.hrmagazine.co.uk/content/features/whatsapp-in-the-workplace, as of December 2, 2022.

[115] W. Luo, Q. Xie, and U. Hengartner, "FaceCloak: An Architecture for User Privacy on Social Networking Sites," in *Conference on Computational Science and Engineering*, ser. CSE '09. Vancouver, Canada: IEEE, Aug. 2009, pp. 26–33.

[116] X. Luo, P. Zhou, E. W. Chan, W. Lee, R. K. Chang, and R. Perdisci, "HTTPOS: Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows," in *Network and Distributed System Security Symposium*, ser. NDSS '11. San Diego, CA, USA: The Internet Society, Feb. 2011.

[117] M. Madejski, M. Johnson, and S. M. Bellovin, "A Study of Privacy Settings Errors in an Online Social Network," in *IEEE International Conference on Pervasive Computing and Communications Workshops*, ser. PerCOM '12. Lugano, Switzerland: IEEE, Mar. 2012, pp. 340–345.

[118] M. J. Mainier, R. Morris, and M. O. Louch, "Social Networks and the Privacy Paradox: A Research Framework," *Issues in Information Systems*, vol. 11, no. 1, pp. 513–517, Nov. 2010.

[119] V. Mayer-Schönberger, *Delete: The Virtue of Forgetting in the Digital Age*. Princeton, NJ, USA: Princeton University Press, Jul. 2011.

[120] M. L. Mazurek, J. Arsenault, J. Breese *et al.*, "Access Control for Home Data Sharing: Attitudes, Needs and Practices," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '10. Atlanta, GA, USA: ACM, Apr. 2010, pp. 645–654.

[121] S. Meredith, "Facebook-Cambridge Analytica: A timeline of the data hijacking scandal," Apr. 2018, https://www.cnbc.com/2018/04/10/facebook-cambridge-analytica-a-timeline-of-the-data-hijacking-scandal.html, as of December 2, 2022.

[122] D. Miller, L. Abed Rabho, P. Wondo, M. de Vries, M. Duque, P. Garvey, L. Haapio-Kirk, C. Hawkins, A. Otaegui, S. Walton, and X. Wang, *The Global Smartphone – Beyond Youth Technology*. London, UK: UCL Press, 2021.

[123] M. Minaei, M. Mondal, P. Loiseau, K. Gummadi, and A. Kate, "Lethe: Conceal Content Deletion from Persistent Observers," in *Privacy Enhancing Technologies Symposium*, ser. PETS '19. Stockholm, Sweden: Sciendo, Jul. 2019, pp. 206–226.

[124] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "SkypeMorph: Protocol Obfuscation for Tor Bridges," in *ACM Conference on Computer and Communications Security*, ser. CCS '12. Raleigh, NC, USA: ACM, Oct. 2012, pp. 97–108.

[125] R. E. Mohamed and S. Chiasson, "Online Privacy and Aging of Digital Artifacts," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '18. Baltimore, MD, USA: USENIX Association, Aug. 2018.

[126] M. Mondal, Y. Liu, B. Viswanath, K. P. Gummadi, and A. Mislove, "Understanding and Specifying Social Access Control Lists," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '14. Menlo Park, CA, USA: USENIX Association, Jul. 2014.

[127] M. Mondal, J. Messias, S. Ghosh, K. P. Gummadi, and A. Kate, "Forgetting in Social Media: Understanding and Controlling Longitudinal Exposure of Socially Shared Data," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '16. Denver, CO, USA: USENIX Association, Jun. 2016, pp. 287–299.

[128] ——, "Longitudinal Privacy Management in Social Media: The Need for Better Controls," *IEEE Internet Computing*, vol. 21, no. 3, pp. 48–55, May 2017.

[129] M. Mondal, G. S. Yilmaz, N. Hirsch, M. T. Khan, M. Tang, C. Tran, C. Kanich, B. Ur, and E. Zheleva, "Moving Beyond Set-It-And-Forget-It Privacy Settings on Social Media," in *ACM Conference on Computer and Communications Security*, ser. CCS '19. London, UK: ACM, Nov. 2019, pp. 991–1008.

[130] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal Adversarial Perturbations," in *IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '17. Honolulu, HI, USA: IEEE, Jul. 2017, pp. 86–94.

[131] G. E. Müller and A. Pilzecker, "Experimentelle Beiträge zur Lehre vom Gedächtniss"," *Zeitschrift für Psychologie und Physiologie der Sinnesorgane*, vol. 1, 1900.

[132] S. J. Murdoch and G. Danezis, "Low-Cost Traffic Analysis of Tor," in *IEEE Symposium on Security and Privacy*, ser. SP '05. Oakland, CA, USA: IEEE, May 2005, pp. 183–195.

[133] S. J. Murdoch and P. Zieliński, "Sampled Traffic Analysis by Internet-Exchange-Level Adversaries," in *Privacy Enhancing Technologies Symposium*, ser. PETS '07. Ottawa, Canada: Springer, Jun. 2007, pp. 167–183.

[134] J. Murdock, "WhatsApp New Feature: Users May Soon Have an Hour to Recall Embarrassing Chats," Mar. 2018, http://www.newsweek.com/whatsapp-users-may-soon-have-hour-delete-embarrassing-chats-groups-830675, as of December 2, 2022.

[135] A. Murillo, A. Kramm, S. Schnorf, and A. De Luca, ""If I press delete, it's gone" - User Understanding of Online Data Deletion and Expiration," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '18. Baltimore, MD, USA: USENIX Association, Aug. 2018, pp. 329–339.

[136] J. Nagy and P. Pecho, "Social Networks Security," in *Conference on Emerging Security Information, Systems and Technologies*, ser. SECUREWARE '09. Athens, Greece: IARIA, Jun. 2009, pp. 321–325.

[137] M. Nasr, A. Bahramali, and A. Houmansadr, "DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning," in *ACM Conference on Computer and Communications Security*, ser. CCS '18. Toronto, Canada: ACM, Oct. 2018, pp. 1962–1976.

[138] M. Nasr, A. Houmansadr, and A. Mazumdar, "Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis," in *ACM Conference on Computer and Communications Security*, ser. CCS '17. Dallas, TX, USA: ACM, Oct. 2017, pp. 2053–2069.

[139] M. Netter, M. Riesner, M. Weber, and G. Pernul, "Privacy Settings in Online Social Networks–Preferences, Perception, and Reality," in *Hawaii International Conference on System Sciences*, ser. HICSS '13. Wailea, HI, USA: IEEE, Jan. 2013, pp. 3219–3228.

[140] C. Niederée, "Learning from Human Memory: Managed Forgetting and Contextualized Remembering for Digital Memories," in *Conference on Theory and Practice of Digital Libraries*, ser. TPDL '15. Poznan, Poland: Springer, Sep. 2015, pp. 1–6.

[141] C. Niederée, N. Kanhabua, F. Gallo, and R. H. Logie, "Forgetful Digital Memory: Towards Brain-inspired Long-term Data and Information Management," *ACM SIGMOD Record*, vol. 44, no. 2, pp. 41–46, Aug. 2015.

[142] R. Nithyanand, O. Starov, A. Zair, P. Gill, and M. Schapira, "Measuring and Mitigating AS-level Adversaries Against Tor," in *Network and Distributed System Security Symposium*, ser. NDSS '16. San Diego, CA, USA: The Internet Society, Feb. 2016.

[143] M. Nouwens, C. F. Griggio, and W. E. Mackay, ""WhatsApp is for family, Messenger is for friends": Communication Places in App Ecosystems," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. Denver, CO, USA: ACM, May 2017, pp. 727–735.

[144] S. E. Oh, T. Yang, N. Mathews, J. K. Holland, M. S. Rahman, N. Hopper, and M. Wright, "DeepCoFFEA: Improved Flow Correlation Attacks on Tor via Metric Learning and Amplification," in *IEEE Symposium on Security and Privacy*, ser. SP '22. San Francisco, CA, USA: IEEE, May 2022.

[145] S. J. Oh, M. Fritz, and B. Schiele, "Adversarial Image Perturbation for Privacy Protection: A Game Theory Perspective," in *IEEE International Conference on Computer Vision*, ser. ICCV '17. Venice, Italy: IEEE, Oct. 2017, pp. 1482–1491.

[146] A.-M. Olteanu, K. Huguenin, I. Dacosta, and J.-P. Hubaux, "Consensual and Privacy-preserving Sharing of Multi-subject and Interdependent Data," in *Network and Distributed System Security Symposium*, ser. NDSS '18. San Diego, CA, USA: The Internet Society, Feb. 2018, pp. 1–16.

[147] Open Whisper Systems, "Signal," May 2010, https://signal.org/, as of December 2, 2022.

[148] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel, "Website Fingerprinting at Internet Scale," in *Network and Distributed System Security Symposium*, ser. NDSS '16. San Diego, CA, USA: The Internet Society, Feb. 2018.

[149] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website Fingerprinting in Onion Routing Based Anonymization Networks," in *Workshop on Privacy in the Electronic Society*, ser. WPES '11. Chicago, IL, USA: ACM, Oct. 2011, pp. 103–114.

[150] K. Park and H. Kim, "Encryption is Not Enough: Inferring User Activities on KakaoTalk with Traffic Analysis," in *International Workshop on Information Security Applications*, ser. WISA '15. Jeju Island, Korea: Springer, Aug. 2015.

[151] Y. J. Park, "Digital Literacy and Privacy Behavior Online," *Communication Research*, vol. 40, no. 2, pp. 215–236, Apr. 2013.

[152] R. Perlman, "The Ephemerizer: Making Data Disappear," *Journal of Information System Security*, vol. 1, no. 1, pp. 51–68, Jan. 2005.

[153] E. Politou, E. Alepis, and C. Patsakis, "Forgetting Personal Data and Revoking Consent Under the GDPR: Challenges and Proposed Solutions," *Journal of Cybersecurity*, vol. 4, no. 1, pp. 1–20, Mar. 2018.

[154] C. Pöpper, D. Basin, S. Capkun, and C. Cremers, "Keeping Data Secret under Full Compromise using Porter Devices," in *ACM Annual Computer Security Applications Conference*, ser. ACSAC '10. Orlando, FL, USA: ACM, Dec. 2010, pp. 241–250.

[155] A. Porter Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS Adoption on the Web," in *USENIX Security Symposium*,

ser. USENIX '17. Baltimore, MD, USA: USENIX Association, Aug. 2018, pp. 1323–1338.

[156] E. Price, "Now WhatsApp Allows You to Delete Messages," Oct. 2017, http://fortune.com/2017/10/27/whatsapp-delete-messages/, as of December 2, 2022.

[157] Proton Technologies AG, "ProtonMail," May 2014, https://protonmail.com/, as of December 2, 2022.

[158] Provable Things Ltd, "Provable - Blockchain Oracle Service, Enabling Data-rich Smart Contracts," 2017, http://provable.xyz, as of December 2, 2022.

[159] M. S. Rahman, M. Imani, N. Mathews, and M. Wright, "Mockingbird: Defending Against Deep-Learning-Based Website Fingerprinting Attacks With Adversarial Traces," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1, pp. 1594–1609, Nov. 2020.

[160] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara, and M. Wright, "Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks," in *Privacy Enhancing Technologies Symposium*, ser. PETS '20. Virtual Event: De Gruyter, Jul. 2020, pp. 5–24.

[161] S. Rajtmajer, A. Squicciarini, J. M. Such, J. Semonsen, and A. Belmonte, "An Ultimatum Game Model for the Evolution of Privacy in Jointly Managed Content," in *Conference on Decision and Game Theory for Security*, ser. GameSec '07. Vienna, Austria: Springer, Oct. 2017, pp. 112–130.

[162] K. M. Ramokapane, A. Rashid, and J. M. Such, ""I Feel Stupid I Can't Delete...": A Study of Users' Cloud Deletion Practices and Coping Strategies," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '17. Santa Clara, CA: USENIX Association, Jul. 2017, pp. 241–256.

[163] Y. Rashidi, T. Ahmed, F. Patel, E. Fath, A. Kapadia, C. Nippert-Eng, and N. M. Su, ""You Don't Want to be the Next Meme": College Students' Workarounds to Manage Privacy in the Era of Pervasive Photography," in *USENIX Symposium On Usable Privacy and Security*, ser. SOUPS '18. Baltimore, MD, USA: USENIX Association, Aug. 2018, pp. 143–157.

[164] J.-F. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," in *Designing Privacy Enhancing Technologies*. Springer, 2001, pp. 10–29.

[165] Real Time Statistics Project, "Internet Live Stats," Oct. 2011, https://www.internetlivestats.com/, as of December 2, 2022.

[166] S. Reimann and M. Dürmuth, "Timed Revocation of User Data: Long Expiration Times from Existing Infrastructure," in *Workshop on Privacy in the Electronic Society*, ser. WPES '12. Raleigh, NC, USA: ACM, Oct. 2012, pp. 65–74.

[167] B. Reynolds, J. Venkatanathan, J. Gonçalves, and V. Kostakos, "Sharing Ephemeral Information in Online Social Networks: Privacy Perceptions and Behaviours," in *IFIP Conference on Human-Computer Interaction*, ser. INTERACT '11. Lisbon, Portugal: IFIP, Sep. 2011, pp. 204–215.

[168] F. Rezaei and A. Houmansadr, "Tagit: Tagging Network Flows Using Blind Fingerprints," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '17. Minneapolis, MN, USA: De Gruyter, Jul. 2017, pp. 290–307.

[169] ——, "FINN: Fingerprinting Network Flows using Neural Networks," in *ACM Annual Computer Security Applications Conference*, ser. ACSAC '21. Online Event: ACM, Dec. 2021, pp. 1011–1024.

[170] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated Website Fingerprinting through Deep Learning," in *Network and Distributed System Security Symposium*, ser. NDSS '18. San Diego, CA, USA: The Internet Society, Feb. 2018.

[171] V. Rimmer, T. Schnitzler, T. Van Goethem, R. Romero, W. Joosen, and K. Kohls, "Trace Oddity: Methodologies for Data-Driven Traffic Analysis on Tor," in *Privacy Enhancing Technologies Symposium*, ser. PETS '22. Sydney, Australia: Sciendo, Jul. 2022, pp. 314–335.

[172] J. Rosen, "The Web Means the End of Forgetting," Jul. 2010, http://archive.nytimes.com/www.nytimes.com/2010/07/25/magazine/25privacy-t2.html, as of December 2, 2022.

[173] A. Schlesinger, E. Chandrasekharan, C. A. Masden, A. S. Bruckman, W. K. Edwards, and R. E. Grinter, "Situated Anonymity: Impacts of Anonymity, Ephemerality, and Hyper-locality on Social Media," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. Denver, CO, USA: ACM, May 2017, pp. 6912–6924.

[174] N. Schmidt, "With swarm data against infection," Mar. 2020, https://www.telekom.com/en/blog/group/article/with-swarm-data-against-infection-597382, as of December 2, 2022.

[175] T. Schnitzler, M. Dürmuth, and C. Pöpper, "Towards Contractual Agreements for Revocation of Online Data," in *IFIP International Conference on ICT*

Systems Security and Privacy Protection, ser. IFIP SEC '19. Lisbon, Portugal: Springer, Jun. 2019, pp. 374–387.

[176] T. Schnitzler, S. Mirza, M. Dürmuth, and C. Pöpper, "SoK: Managing Longitudinal Privacy of Publicly Shared Personal Online Data," in *Privacy Enhancing Technologies Symposium*, ser. PETS '21. Virtual Event: Sciendo, Jan. 2021, pp. 229–249.

[177] T. Schnitzler, C. Pöpper, M. Dürmuth, and K. Kohls, "We Built This Circuit: Exploring Threat Vectors in Circuit Establishment in Tor," in *IEEE European Symposium on Security and Privacy*, ser. EuroS&P '21. Virtual Event: IEEE, Sep. 2021, pp. 319–336.

[178] T. Schnitzler, C. Utz, F. M. Farke, C. Pöpper, and M. Dürmuth, "User Perception and Expectations on Deleting Instant Messages – or – "What Happens If I Press This Button?"," in *European Workshop on Usable Security*, ser. EuroUSEC '18. London, UK: The Internet Society, Apr. 2018, pp. 1–9.

[179] T. Schnitzler, C. Utz, F. M. Farke, C. Pöpper, and M. Dürmuth, "Exploring User Perceptions of Deletion in Mobile Instant Messaging Applications," *Journal of Cybersecurity*, vol. 6, no. 1, pp. 1–15, Jan. 2020.

[180] H. Sengar, Z. Ren, H. Wang, D. Wijesekera, and S. Jajodia, "Tracking Skype Voip Calls Over the Internet," in *IEEE International Conference on Computer Communications*, ser. INFOCOM '10. San Diego, CA, USA: IEEE, Jul. 2010, pp. 1–5.

[181] A. Serjantov and P. Sewell, "Passive Attack Analysis for Connection-Based Anonymity Systems," in *European Symposium on Research in Computer Security*, ser. ESORICS '03. Gjøvik, Norway: Springer, Oct. 2003, pp. 116–131.

[182] A. Shah, R. Fontugne, and C. Papadopoulos, "Towards Characterizing International Routing Detours," in *Asian Internet Engineering Conference*, ser. AINTEC '16. Bangkok, Thailand: ACM, Nov. 2016, pp. 17–24.

[183] S. Sharwood, "AWS US East region endures eight-hour wobble thanks to 'Stuck IO' in Elastic Block Store," Sep. 2021, https://www.theregister.com/2021/09/28/aws_east_brownout/, as of December 2, 2022.

[184] E. Shein, "Ephemeral Data," *Communications of the ACM*, vol. 56, no. 9, pp. 20–22, Sep. 2013.

[185] V. Shmatikov and M.-H. Wang, "Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses," in *European Symposium on Research in Computer Security*, ser. ESORICS '06. Hamburg, Germany: Springer, Sep. 2006, pp. 18–33.

[186] S. Singanamalla, E. H. B. Jang, R. Anderson, T. Kohno, and K. Heimerl, "Accept the Risk and Continue: Measuring the Long Tail of Government https Adoption," in *ACM Internet Measurement Conference*, ser. IMC '20. Virtual Event: ACM, Oct. 2020, pp. 577–597.

[187] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning," in *ACM Conference on Computer and Communications Security*, ser. CCS '18. Toronto, Canada: ACM, Oct. 2018, pp. 1928–1943.

[188] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with n-Shot Learning," in *ACM Conference on Computer and Communications Security*, ser. CCS '19. London, UK: ACM, Nov. 2019, pp. 1131–1148.

[189] C. Sitawarin, A. N. Bhagoji, A. Mosenia, P. Mittal, and M. Chiang, "Rogue Signs: Deceiving Traffic Sign Recognition with Malicious Ads and Logos," in *IEEE Deep Learning and Security Workshop*, ser. DLS '18. San Francisco, CA, USA: IEEE, May 2018.

[190] M. Sleeper, J. Cranshaw, P. G. Kelley, B. Ur, A. Acquisti, L. F. Cranor, and N. Sadeh, ""I Read My Twitter the Next Morning and Was Astonished": A Conversational Perspective on Twitter Regrets," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '13. Paris, France: ACM, Apr. 2013, pp. 3277–3286.

[191] M. Sleeper, W. Melicher, H. Habib, L. Bauer, L. F. Cranor, and M. L. Mazurek, "Sharing Personal Content Online: Exploring Channel Choice and Multi-Channel Behaviors," in *ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. Santa Clara, CA, USA: ACM, May 2016, pp. 101–112.

[192] Snap Inc., "Snapchat," Sep. 2011, https://www.snapchat.com/, as of December 2, 2022.

[193] P. Snyder and C. Kanich, "Cloudsweeper: Enabling Data-centric Document Management for Secure Cloud Archives," in *ACM Cloud Computing Security Workshop*, ser. CCSW '13. Berlin, Germany: ACM, Nov. 2013, pp. 47–54.

[194] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 477–560, Jan. 2006.

[195] A. C. Squicciarini, M. Shehab, and F. Paci, "Collective Privacy Management in Social Networks," in *The Web Conference*, ser. WWW '09. Madrid, Spain: ACM, Apr. 2009, pp. 521–530.

[196] Statista Inc., "Number of Mobile Phone Messaging App Users Worldwide From 2016 to 2021 (in Billions)," Jul. 2017, https://www.statista.com/statistics/483255/number-of-mobile-messaging-users-worldwide/, as of December 2, 2022.

[197] ——, "Most Popular Global Mobile Messenger Apps as of October 2018, Based on Number of Monthly Active Users (in Millions)," Oct. 2018, https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, as of December 2, 2022.

[198] ——, "Most Popular Mobile Messaging Apps in the United States as of July 2018, by Monthly Active Users (in Millions)," Jul. 2018, https://www.statista.com/statistics/350461/mobile-messenger-app-usage-usa/, as of December 2, 2022.

[199] ——, "Number of Global Monthly Active Kakaotalk Users from 1st Quarter 2013 to 3rd Quarter 2018 (in Millions)," Oct. 2018, https://www.statista.com/statistics/278846/kakaotalk-monthly-active-users-mau/, as of December 2, 2022.

[200] ——, "Distribution of Facebook users worldwide as of January 2020, by age and gender," Feb. 2020, https://www.statista.com/statistics/376128/facebook-global-user-age-distribution/, as of December 2, 2022.

[201] ——, "Number of Internet Users Worldwide From 2005 to 2019 ," Nov. 2020, https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/, as of December 2, 2022.

[202] ——, "Most Popular Global Mobile Messenger Apps as of July 2021, Based on Number of Monthly Active Users ," Jul. 2021, https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, as of December 2, 2022.

[203] ——, "Most Popular Social Networks Worldwide as of October 2021, Ranked by Number of Active Users ," Oct. 2021, https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/, as of December 2, 2022.

[204] ——, "Most Popular Global Mobile Messenger Apps as of January 2022, Based on Number of Monthly Active Users ," Jan. 2022, https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, as of December 2, 2022.

[205] K. Stokes and N. Carlsson, "A Peer-to-Peer Agent Community for Digital Oblivion in Online Social Networks," in *IEEE Conference on Privacy, Security and Trust*, ser. PST '13. Tarragona, Spain: IEEE, Jul. 2013, pp. 103–110.

[206] F. D. Stutzman, R. Gross, and A. Acquisti, "Silent Listeners: The Evolution of Privacy and Disclosure on Facebook," *Journal of Privacy and Confidentiality*, vol. 4, no. 2, pp. 7–41, Mar. 2013.

[207] J. M. Such and N. Criado, "Resolving Multi-Party Privacy Conflicts in Social Media," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1851–1863, Jul. 2016.

[208] A. Sulleyman, "WhatsApp's 'Delete for Everyone' Feature Lets You Unsend Embarrassing Messages: Here's How to Use it," Oct. 2017, https://www.independent.co.uk/life-style/gadgets-and-tech/news/whatsapp-delete-unsend-message-for-everyone-before-read-how-to-use-how-does-it-work-a8022816.html, as of December 2, 2022.

[209] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical Identification of Encrypted Web Browsing Traffic," in *IEEE Symposium on Security and Privacy*, ser. SP '02. Berkeley, CA, USA: IEEE, May 2002, pp. 19–30.

[210] Y. Sun, A. Edmundson, N. Feamster, M. Chiang, and P. Mittal, "Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks," in *IEEE Symposium on Security and Privacy*, ser. S&P '17. San Jose, CA, USA: IEEE, May 2017, pp. 977–992.

[211] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "RAPTOR: Routing Attacks on Privacy in Tor," in *USENIX Security Symposium*, ser. USENIX '16. Austin, TX, USA: USENIX Association, Aug. 2016, pp. 271–286.

[212] P. Syverson, R. Dingledine, and N. Mathewson, "Tor: The Second-Generation Onion Router," in *USENIX Security Symposium*, ser. USENIX '04. Boston, MA, USA: USENIX Association, Jun. 2004.

[213] H. Tan, M. Sherr, and W. Zhou, "Data-Plane Defenses Against Routing Attacks on Tor," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '16. Darmstadt, Germany: De Gruyter, Oct. 2016, pp. 276–293.

[214] Telegram Messenger LLP, "Telegram," Aug. 2013, https://telegram.org/, as of December 2, 2022.

[215] The Tor Project, Inc., "Tor Metrics," 2009, https://metrics.torproject.org/, as of December 2, 2022.

[216] K. Thomas, C. Grier, and D. M. Nicol, "unFriendly: Multi-party Privacy Risks in Social Networks," in *Privacy Enhancing Technologies Symposium*, ser. PETS '10. Berlin, Germany: Springer, Jul. 2010, pp. 236–252.

[217] Threema GmbH, "Threema," Dec. 2012, https://www.threema.ch/, as of December 2, 2022.

[218] ——, "Security and Privacy FAQ," 2020, https://threema.ch/en/security, as of December 2, 2022.

[219] M. Trevisan, D. Giordano, I. Drago, M. M. Munafò, and M. Mellia, "Five Years at the Edge: Watching Internet From the ISP Network," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 561–574, Apr. 2020.

[220] J. Y. Tsai, S. Egelman, L. F. Cranor, and A. Acquisti, "The Effect of Online Privacy Information on Purchasing Behavior: An Experimental Study," in *International Conference on Information Systems*, ser. ICIS '07. Montreal, QC, Canada: Association for Information Systems, Dec. 2007, pp. 254–268.

[221] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith, "SoK: Secure Messaging," in *IEEE Symposium on Security and Privacy*, ser. SP '15. San Jose, CA, USA: IEEE, May 2015, pp. 232–249.

[222] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philosophical Transactions of the Royal Society A*, vol. 376, no. 2133, pp. 1–15, Jul. 2018.

[223] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," in *IEEE International Conference on Image Processing*, ser. ICIP '00. Vancouver, Canada: IEEE, Aug. 2000, pp. 664–666.

[224] E. F. Villaronga, P. Kieseberg, and T. Li, "Humans Forget, Machines Remember: Artificial Intelligence and the Right to be Forgotten," *Computer Security& Law Review*, vol. 34, no. 2, pp. 1–19, Aug. 2017.

[225] G. Wan, A. Johnson, R. Wails, S. Wagh, and P. Mittal, "Guard Placement Attacks on Path Selection Algorithms for Tor," in *Privacy Enhancing Technologies Symposium*, ser. PoPETS '19. Stockholm, Sweden: Sciendo, Jul. 2019, pp. 272–291.

[226] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective Attacks and Provable Defenses for Website Fingerprinting," in *USENIX Security Symposium*, ser. USENIX '14. San Diego, CA, USA: USENIX Association, Aug. 2014, pp. 271–286.

[227] T. Wang and I. Goldberg, "Improved Website Fingerprinting on Tor," in *Workshop on Privacy in the Electronic Society*, ser. WPES '13. Berlin, Germany: ACM, Nov. 2013.

[228] ——, "On Realistically Attacking Tor with Website Fingerprinting," in *Privacy Enhancing Technologies Symposium*, ser. PETS '16. Darmstadt, Germany: De Gruyter, Jul. 2016, pp. 21–36.

[229] ——, "Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks," in *USENIX Security Symposium*, ser. USENIX '17. Vancouver, Canada: USENIX Association, Aug. 2017, pp. 1375–1390.

[230] X. Wang, S. Chen, and S. Jajodia, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," in *IEEE Symposium on Security and Privacy*, ser. SP '07. Oakland, CA, USA: IEEE, May 2007, pp. 116–130.

[231] X. Wang, D. S. Reeves, and F. Wu, "Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones," in *European Symposium on Research in Computer Security*, ser. ESORICS '02. Zurich, Switzerland: Springer, Sep. 2002, pp. 244–263.

[232] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor, ""I Regretted the Minute I Pressed Share": A Qualitative Study of Regrets on Facebook," in *ACM Symposium On Usable Privacy and Security*, ser. SOUPS '11. Pittsburgh, PA, USA: ACM, Jul. 2011, pp. 1–16.

[233] Z. Wang, "The Applications of Deep Learning on Traffic Identification," in *Black Hat Technical Security Conference*, ser. BLACKHAT '15, Las Vegas, NV, USA, Aug. 2015.

[234] R. Wash, "Folk Models of Home Computer Security," in *ACM Symposium On Usable Privacy and Security*, ser. SOUPS '10. Redmond, WA, USA: ACM, Jul. 2010.

[235] G. Wegberg, H. Ritzdorf, and S. Capkun, "Multi-User Secure Deletion on Agnostic Cloud Storage," ETH Zurich, Tech. Rep., 2017.

[236] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, "StegoTorus: A Camouflage Proxy for the Tor Anonymity System," in *ACM Conference on Computer and Communications Security*, ser. CCS '12. Raleigh, NC, USA: ACM, Oct. 2012, pp. 109–120.

[237] WhatsApp LLC, "How to check read receipts," https://faq.whatsapp.com/android/security-and-privacy/how-to-check-read-receipts/, as of December 2, 2022.

[238] ——, "WhatsApp," Jan. 2009, https://www.whatsapp.com/, as of December 2, 2022.

[239] A. Whitten and J. D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," in *USENIX Security Symposium*, ser. SSYM '99. Berkeley, CA, USA: USENIX Association, Aug. 1999, pp. 14–14.

[240] M. Williams, "Secure Messaging Apps Comparison," 2021, https://www.securemessagingapps.com/, as of December 2, 2022.

[241] R. Wishart, D. Corapi, S. Marinovic, and M. Sloman, "Collaborative Privacy Policy Authoring in a Social Networking Context," in *Symposium on Policies for Distributed Systems and Networks*, ser. POLICY '10. Washington D. C., USA: IEEE, Jul. 2010, pp. 1–8.

[242] B. Wolford, "Everything you need to know about the "Right to be forgotten"," https://gdpr.eu/right-to-be-forgotten/, as of December 2, 2022.

[243] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot Me if You Can: Uncovering Spoken Phrases in Encrypted VoIP Conversations," in *IEEE Symposium on Security and Privacy*, ser. S&P '08. Oakland, CA, USA: IEEE, May 2008, pp. 35–49.

[244] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob?" in *USENIX Security Symposium*, ser. USENIX '07. Boston, MA, USA: USENIX Association, Aug. 2007, pp. 43–54.

[245] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," in *Network and Distributed System Security Symposium*, ser. NDSS '09. San Diego, CA, USA: The Internet Society, Feb. 2009.

[246] Y. Xue, K. Xue, N. Gai, J. Hong, D. S. L. Wei, and P. Hong, "An Attribute-Based Controlled Collaborative Access Control Scheme for Public Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 2927–2942, Nov. 2019.

[247] B. Yang, F. Gu, and X. Niu, "Block Mean Value Based Image Perceptual Hashing," in *Conference on Intelligent Information Hiding and Multimedia Signal Processing*, ser. IIH-MSP '06. Pasadena, CA, USA: IEEE, Dec. 2006, pp. 167–172.

[248] C. W. Yoo, H. J. Ahn, and H. R. Rao, "An Exploration of the Impact of Information Privacy Invasion," in *International Conference on Information Systems*, ser. ICIS '12. Orlando, FL, USA: Association for Information Systems, Dec. 2012.

[249] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "DSSS-Based Flow Marking Technique for Invisible Traceback," in *IEEE Symposium on Security and Privacy*, ser. SP '07. Oakland, CA, USA: IEEE, May 2007, pp. 18–32.

[250] A. Zarras, K. Kohls, M. Dürmuth, and C. Pöpper, "Neuralyzer: Flexible Expiration Times for the Revocation of Online Data," in *ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '16. New Orleans, LA, USA: ACM, Mar. 2016, pp. 14–25.

[251] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town Crier: An Authenticated Data Feed for Smart Contracts," in *ACM Conference on Computer and Communications Security*, ser. CCS '16. Vienna, Austria: ACM, Oct. 2016, pp. 270–282.

[252] L. Zhou, W. Wang, and K. Chen, "Tweet Properly: Analyzing Deleted Tweets to Understand and Identify Regrettable Ones," in *The Web Conference*, ser. WWW '16. Montreal, Canada: ACM, Apr. 2016, pp. 603–612.

[253] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On Flow Correlation Attacks and Countermeasures in Mix Networks," in *Privacy Enhancing Technologies Workshop*, ser. PET '04. Toronto, Canada: Springer, May 2004, pp. 207–225.