LEONIE REICHERT

PRIVACY-PRESERVING DATA ANALYSIS AND DISTRIBUTED PROCESSING IN PANDEMIC SETTINGS AND BEYOND

PRIVACY-PRESERVING DATA ANALYSIS AND DISTRIBUTED PROCESSING IN PANDEMIC SETTINGS AND BEYOND

DISSERTATION

von

Leonie Reichert

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)
im Fach Informatik

genehmigt im
Fachbereich Informatik
der Technischen Universität Darmstadt

Präsidentin der Technischen Universität Darmstadt
Prof. Dr. Tanja Brühl

Dekan des Fachbereichs Informatik
Prof. Dr. Dr. Christian Reuter

1. Gutachter: Prof. Dr. Björn Scheuermann
2. Gutachter: Dr. Wouter Lueks

Darmstadt 2024

Leonie Reichert: *Privacy-Preserving Data Analysis and Distributed Processing in Pandemic Settings and Beyond*, A Dissertation on Privacy by Design,
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2024,
URN: urn:nbn:de:tuda-tuprints-285959
Tag der mündlichen Prüfung: 08.10.2024

ABSTRACT

Privacy is acknowledged as a fundamental human right and essential for the functioning of modern democracies, particularly as research and the economy become increasingly data driven. The Covid-19 pandemic has given rise to many new applications necessitating the processing of sensitive information such as health, location, and proximity data. Notable examples include discovering new infections by retracing the contacts of diagnosed individuals and identifying super-spreader events through presence tracing. To fight a pandemic, gaining meaningful statistical insight on the current epidemiological situation based on health data – or other sensitive information – is important. For all these applications, the processed data can reveal private information regarding the data providers. Therefore, data providers require concrete privacy guarantees at every step.

This thesis focuses on solutions for processing and analyzing sensitive data in a privacy-preserving way without requiring trust being place in a central authority. Multiple approaches are proposed to ensure privacy during distributed data processing for Digital Contact Tracing (DCT). To establish a general understanding of the topic, an introduction to the problems and solutions for DCT is presented. The literature is systematized and common challenges with regard to privacy, security, and functionality are identified. Based on the shortcomings of existing contact tracing applications, novel designs for privacy-preserving DCT are presented, along with their respective advantages and drawbacks. The focus is on distributing the tracing process and risk-scoring tasks to users while mitigating the leakage of private data through metadata. Strong privacy guarantees are also provided by using cryptographic primitives such as blind signatures, Oblivious Random Access Memory (ORAM), and Private Set Intersection (PSI). Such techniques allow the design of protocols that only reveal the minimal required amount of information to all parties involved. Systems for super-spreader detection through presence tracing are also presented that can be integrated with DCT systems in a privacy-preserving manner.

While decentralized processing provides better privacy than the centralized alternative, it limits the ability to observe the epidemic situation through statistical analysis. By reviewing common approaches for collecting and analyzing health data for research purposes, we identify various threats to the privacy of people who are willing to share their data. Both in the pandemic and post-pandemic settings, privacy guarantees are a tool to ensure to data providers that their data can not be misused. To this end, a platform is presented that leverages Trusted Execution Environments (TEEs) in combination with oblivious algo-

rithms that safeguard sensitive data during data collection and analysis. To combat the drawbacks of TEEs, new methods are introduced to hide the access patterns and volume patterns of database queries. All contributions presented in this thesis aim to improve the privacy of individuals through solutions that follow the concept of privacy by design.

ZUSAMMENFASSUNG

Der Schutz der Privatsphäre ist ein Menschenrecht und ein unerlässlicher Bestandteil von moderne Demokratien. Da Forschung und vor allem die Wirtschaft zunehmend datengesteuert sind, gewinnt dieser Aspekt weiter an Bedeutung. Die Covid-19 Pandemie hat viele neue Anwendungen hervorgebracht, welche die Verarbeitung sensibler Informationen zu Gesundheit, Standort und sozialen Interaktionen erfordern. Die digitale Kontaktnachverfolgung durch Abstandsbestimmung und die Erkennung von Super-Spreader Ereignissen an öffentlichen Orten sind Beispiele für solche Anwendungen. Zum Bekämpfen einer Pandemie ist es nicht nur relevant neue Infektionen zu verhindern. Um Entscheidungen treffen zu können, benötigt es einen repräsentativen Einblick in das momentane Infektionsgeschehen. Für diesen Zweck muss auf der Grundlage von Gesundheitsdaten und anderen sensiblen Informationen aussagekräftige statistische Erkenntnisse gewonnen werden können. Für all diese Anwendungen benötigen die Personen, welche ihre Daten freiwillig bereitstellen, konkrete Datenschutzgarantien,um sicher sein zu können, dass ihre Daten nicht zweckentfremdet werden oder abhanden kommen. Diese Doktorarbeit konzentriert sich auf Lösungen für die Verarbeitung und Analyse sensibler Daten unter Wahrung der Privatsphäre. Der Schwerpunkt liegt hier auf der Entwicklung von Systemen und Algorithmen, bei denen kein blindes Vertrauen in einer zentralen Autorität gesetzt werden muss, sonder Datenschutz auf andere Weise garantiert werden kann.

In dieser Doktorarbeit werden mehrere Ansätze zur Sicherstellung des Datenschutzes bei der digitale Kontaktverfolgung – im Englischen *Digital Contact Tracing* (DCT) – vorgeschlagen. Um ein allgemeines Verständnis für das Thema zu schaffen, wird eine Einführung in die Probleme und Lösungen für digitale Kontaktnachverfolgung mittels Bluetooth Low Energy (BLE) zur Abstandsbestimmung gegeben. Auf Basis der Literatur wird eine Systematisierung der verschiedenen Ansätze erarbeitet, anhand welcher sich gemeinsame Herausforderungen in Bezug auf Datenschutz, Sicherheit und Funktionalität identifizieren lassen. Ausgehend von den Unzulänglichkeiten bestehender Anwendungen zur Kontaktverfolgung werden neuartige Entwürfe für datenschutzfreundliche digitale Kontaktnachverfolgung mit ihren jeweiligen Vor- und Nachteilen vorgestellt. Ein Schwerpunkt liegt dabei auf der Verteilung der Risikobewertung an die Benutzer, welche durch anonyme Direktnachrichten ihre Kontakte bezüglich möglicher Infektionsrisiken warnen. Zu diesem Zweck werden Fragen der Authentizität von Warnungen angegangen und der Verlust von Privatsphäre durch Metadaten eingedämmt. Noch stärkere Datenschutzgarantien werden auch durch

die Verwendung kryptographischer Protokolle wie Oblivious Random Access Memory (ORAM) und Private Set Intersection (PSI) für digitale Kontaktnachverfolgung erreicht. Derartig Techniken ermöglichen Protokollen, bei denen jede Partei nicht mehr sensible Informationen erhält als zwingend notwendig. Auch werden Möglichkeiten angesprochen, wie digitale Kontaktnachverfolgung für Super-Spreader Erkennung in existierende abstandsbasierte Systeme integriert werden kann auf eine Privatsphäre-erhaltende Weise.

Eine dezentrale Risikobewertung für Kontaktnachverfolgung, wie etwa bei der Corona Warn-App, bietet zwar einen besseren Schutz der Privatsphäre als die zentralisierte Alternative, schränkt aber die Möglichkeit ein, die epidemische Situation durch statistische Auswertungen zu beobachten. Bei der Überprüfung gängiger Ansätze für die Erhebung und Analyse von Gesundheitsdaten durch mobile Geräte zu Forschungszwecken stellen wir verschiedene Gefahren für die Privatsphäre von Menschen fest, welche bereit sind, ihre Daten zu teilen. Sowohl in der Pandemie, als auch unabhängig davon, sind Datenschutzgarantien ein Instrument, mit dem sichergestellt werden kann, dass die Daten von Freiwilligen nicht missbraucht und zweckentfremdet werden können. Zu diesem Zweck wird eine Plattform vorgestellt, welche Trusted Execution Environments (TEEs) verwendet, um Datenanalysen auf sensiblen Daten zu realisieren. Um verschiedene Schwächen von TEEs auszugleichen, werden spezielle Algorithmen vorgestellt, deren Ziel es ist zu Verhindern, dass ein Angreifer lernt auf welche Daten zugegriffen und wie viele Daten für eine Datenbankanfrage verarbeitet werden. Alle vorgestellten Beiträge haben das Ziel, die Daten von Individuen besser zu schützen.

PREVIOUSLY PUBLISHED MATERIAL AND USED TOOLS

This thesis includes material published previously in scientific journals and conferences. This section explains how these publications map to chapters in this thesis. As scientific work is usually the result of a group effort, the following paragraphs also clarify the individual contributions of my co-authors and me. Affiliations at the time of publication are stated once for each work. Unless noted otherwise, my co-authors and I were affiliated at the Humboldt Universität zu Berlin. For all publications, Björn Scheuermann provided general supervision and feedback on the written parts.

Chapter 3 provides an overview of proximity-based approaches to Digital Contact Tracing (DCT) and is based on a survey written together with Samuel Brack and Björn Scheuermann [2]. The survey was published in the *ACM Transactions on Computing for Healthcare* in 2021. Collecting references and systemizing publications on DCT, as well as writing the survey, was primarily done by myself. Samuel Brack provided support during the literature review and writing. The contents of the survey were split into multiple parts and reorganized for this thesis. Portions, especially the discussion in Section 3.7, had to be rewritten to reflect the current state of knowledge in 2024. Chapter 3 contains the overview of proximity-based DCT, while other parts of the work were used as related work in Chapter 4 and Chapter 5.

Chapter 4 discusses two approaches to client-side DCT with direct messaging. Section 4.2 presents the design CAUDHT and is based on a paper written together with Samuel Brack (first author) and Björn Scheuermann [1]. It was published at the *Conference on Local Computer Networks* (*LCN*) in 2020. The concept of CAUDHT was developed jointly with Samuel Brack. Using blind signatures to provide authenticity of warnings in a distributed setting was a designated contribution of Samuel Brack. The idea of leveraging a Distributed Hash Table (DHT) instead of relying on a potentially malicious messaging server originated from me. For the text of this thesis, additional considerations were added regarding the use of Elliptic Curve Cryptography (ECC) and the compression of such keys. All authors contributed to the text of the paper.

Section 4.3 in the same chapter builds on CAUDHT and is based on previous work created in collaboration with Samuel Brack and Björn Scheurmann [4]. The paper was presented at the *Workshop on Secure IT Technologies against Covid-19* (*CoronaDef*) of the *Network and Distributed Systems Security* (*NDSS*) *Symposium* in 2021. Conception and evaluation were done in collaboration with Samuel Brack. The idea of using ring

signatures, as well as the code for computing blind signatures, were distinct contributions of Samuel Brack. All authors contributed to the text of the paper.

Chapter 5 looks at solutions using cryptographic protocols for DCT. Section 5.3 discusses DCT using an Oblivious Random Access Memory (ORAM). It is based on a poster presented at the poster session of the *IEEE Symposium on Security and Privacy* (*S&P*) in March 2020 [5]. The poster was the result of joint work with Samuel Brack and Björn Scheuermann. The idea for this poster, as well as the code and evaluation, originated from me. Samuel Brack and Björn Scheuermann assisted during the conception phase and the writing process. The evaluation was extended for this thesis.

The paper discussed in Section 5.4 uses circuit-based Private Set Intersection (PSI) for DCT. It was published in collaboration with Marcel Pazelt and Björn Scheuermann at the *IEEE International Performance, Computing, and Communications Conference* (*IPCCC*) in 2021 [7]. It is based on the Master thesis of Marcel Pazelt, which was written under my supervision and assistance. Marcel Pazelt contributed a review of the related literature, as well as the implementation and evaluation of the final design. The idea, as well as the largest parts of the text of the paper, originated from me.

Chapter 6 presents an approach to presence tracing on top of the Google Apple Exposure Notification (GAEN) framework. It is based on joint work with Samuel Brack and Björn Scheuermann, which was presented at the *Workshop on Communication, IoT, and AI Technologies to Counter Covid-19* (*COVI-COM*) at the *International Conference on Communications* (*ICC*) in 2021. The idea and design originated from me. Samuel Brack assisted in reviewing the relevant literature and writing.

Chapter 7 is based on a paper written together with Björn Scheuermann (Technical University Darmstadt) and presented at the *International Workshop on Privacy Engineering* at the *IEEE European Symposium on Security and Privacy* (*Euro S&P*) in 2023. The idea, literature review, and security analysis were all done by me. Björn Scheuermann supervised during this process.

Chapter 8 is based on a paper accepted at the *ACM ASIA Conference on Computer and Communications Security* (*ASIA CCS*) 2024. It was written in collaboration with Gowri R Chandran (Technical University Darmstadt), Phillipp Schoppmann (Google), Thomas Schneider (Technical University Darmstadt), and Björn Scheuermann (Technical University Darmstadt). The idea, the conception, the implementation, and the evaluation originated from me. Phillipp Schoppmann and Gowri R Chandran contributed to the theoretical proofs and threat model. Using a truncated Laplace function is a distinct contribution of Phillipp Schoppmann. The text of the paper was mainly written by myself with assistance from Gowri R Chandran and Phillipp Schoppmann. Thomas Schneider provided supervision and proofread the text. At the time

of publication of this paper, I am affiliated at the Technical University Darmstadt. However, a large part of the work on this paper was done during my time at the Humboldt Universität zu Berlin.

This thesis is the result of independent work. In parts, it has been linguistically revised with the help of the tools Grammarly, DeepL and ChatGPT (version 3.5). DeepL was used to translate content written by myself from German to English. Grammerly was used to ensure proper used to correct grammar, spelling, and phrasing. ChatGPT was used in Chapter 1 und 8 to alter text sections written by myself to sound more fluent and natural. These generated outputs were only used as suggestion for improvements. All tools that rely on artificial intelligence in some form were employed solely for the purpose of revising grammatical structure and ensuring clarity of expression.

ACKNOWLEDGEMENTS

First of all, I would like to thank Björn Scheuermann for his supervision and the opportunity to pursue the topic of *Privacy by Design* in my PhD thesis, first at his chair at the Humboldt University of Berlin and from January 2024 onward at the Technical University of Darmstadt.

This PhD thesis would not have been possible without Samuel Brack. Thank you for the numerous hours in Zoom calls and for making the Covid-19 pandemic a pleasant, entertaining, and productive time.

I am also grateful to Florian Tschorsch for his mentorship, support, and invaluable advice, and for taking the time to talk about various questions and problems.

Special thanks go to Martin Florian, whose suggestions on various papers and readiness to listen, even on non-research matters, have been greatly appreciated.

I also thank my other co-authors: Phillipp Schoppmann for his great insights, Gowri R Chandran for assistance with mathematical proofs, and Thomas Schneider for sharing his extensive knowledge of cryptographic protocols.

During my dissertation, I supervised several Master's and Bachelor's theses that contributed to this work in various ways. Special thanks go to Marcel Pazelt, Alexander Brunkow, and Alexander Senger.

My appreciation goes out to the technical and administrative staff at both the Humboldt University and the Technical University Darmstadt for making life as PhD student easier. I am also grateful to the team at the Weizenbaum Institute and the colleagues from the 5Genesis and 5G Victori research projects. Of course, the collegues and fellow PhD students at the former Chair of Technical Computer Science ("Lehrstuhl Technische Informatik") of the Humboldt University and at KOM (both the research group for Communication Networks as well as the Multi-Media Communications Lab) at the Technical University Darmstadt should not be missing from this list. Many thanks to everyone for the great time!

An important contribution to the readability of this document came from all those who were kind enough to read through it in whole or in parts. Many thanks to Jan Götte, Samuel Brack, Nicole Vieregg, Sebastian Rust, Florian Tschorsch, and Björn Scheuermann.

Finally, I would like to thank my friends and companions who have enriched the last five years. Special thanks also go to my family, who have given me the opportunity to choose my own path.

CONTENTS

ACRONYMS

BLE	Bluetooth Low Energy
DCT	Digital Contact Tracing
(D)DoS	(Distributed) Denial-of-Service
DHT	Distributed Hash Table
DORAM	Doubly-Oblivious Random Access Memory
DOSM	Doubly-Oblivious Sorted Multimap
DP	Differential Privacy
ECC	Elliptic Curve Cryptography
GAEN	Google Apple Exposure Notifications
GDPR	General Data Protection Regulation
IPFS	Interplanetary File System
MPC	(Secure) Multi-Party Computation
ODB	Oblivious Database
OPPRF	Oblivious Programmable Pseudo-Random Functions
ORAM	Oblivious Random Access Memory
PKI	Public Key Infrastructure
PSI	Private Set Intersection
TEE	Trusted Execution Environment

Part I

PROLOGUE

1 INTRODUCTION

1.1 MOTIVATION

Privacy is a common good and is widely acknowledged as a basic human right, as emphasized by the Universal Declaration of Human Rights Article 12 [308]. It is a requirement for the protection of fundamental values such as personal autonomy, individuality, and dignity. The loss of privacy, or sensitive data in the wrong hands, can have serious consequences and negatively impact the private lives of those concerned. Health data is especially vulnerable as it can contain information that can be cause for stigmatization, discrimination, or even exploitation.

However, large parts of the economy of the 21st century are data driven and personal information is used for various purposes. Data is commonly processed and stored in a centralized manner without technical measures for privacy protection, relying only on legal frameworks. Such centralized data processing requires data sources to trust the processing entity as it makes decisions on its further use from this point onwards. If a central data processing entity was benign at the beginning, data might still be misused at a later point in time as such methodologies are prone to forced access by third parties like law enforcement or hackers.

To minimize potential harm to individuals, it is necessary to move away from models where privacy is only provided through mutual trust assumptions and towards privacy by design. To this end, algorithms that inherently protect privacy through provable guarantees are crucial to harness the power of sensitive data sources responsibly. Such approaches foster innovation and facilitate research in fields like medicine that depend on such data. The Covid-19 pandemic serves as a great example of the versatility of privacy-preserving technologies as it catalyzed advancements in various domains. New challenges required processing large amounts of private data regarding health status, location history, and social interactions. Privacy guarantees became a tool to ensure the cooperation of large parts of the population.

This thesis examines how privacy can be preserved in data processing and analysis without placing trust in a central entity. We look at applications related to the Covid-19 pandemic and provide solutions for utilizing health, location, and proximity data in a manner that adheres to the principle of privacy by design. Some findings are also applicable in more general, post-pandemic settings. To gain statistical insight into such data, we also provide privacy-preserving methodologies for analyzing sensitive data.

1.1.1 *Challenges*

During the pandemic, many new ideas to digitalize the fight against Covid-19 were pitched, tested, and rolled out on a grand scale in a short time. The applications ranged from symptom checkers, quarantine enforcement, and health certificates to contact tracing [189]. The latter is a method for handling infection chains and aims to analyze the social contacts of diagnosed people to discover new infections as early as possible. It was, therefore, an essential tool in combating the Covid-19 pandemic. Digitalizing and automating the process, however, requires large amounts of highly sensitive information regarding who interacted with whom, not only from infected people but also from those who are still healthy.

Gaining the trust of large parts of the population is the key to ensuring their participation in *Digital Contact Tracing* (DCT) [189]. Large adoption rates of voluntary contact tracing apps are directly linked to an improvement in the overall effectiveness of these systems [157]. To this end, users' fear of data misuse needs to be answered in a reliable and credible manner. The establishing of surveillance measures for the purpose of contact tracing, as done by Singapore, South Korea, and Israel at the wake of the pandemic [147, 149, 289], is widely regarded as a threat to democracy and as a source of a loss of trust in the government [19, 115, 195]. Methods for addressing these challenges encompass an architectural shift in the design of proximity-based contact tracing. This change ensures that users are not required to trust in the benevolence of a central authority, such as the government or a health authority, as long as they can be sure that the system works as intended. As seen also in the case of the Covid-19 pandemic, once data is stored in a centralized manner, it will be used by law enforcement for other purposes independent of prior promises [71, 218, 260].

Although the distribution of risk detection to clients provides privacy to those wanting to determine their infection risk, it leaves room for deanonymization attacks against diagnosed people. Client-side risk assessment leaks information to users who have received a warning regarding the exact or approximate time when they encountered a diagnosed person. This information can be used to deanonymize the source of a warning, especially if more than one encounter occurred.

In an epidemic or pandemic, so-called super-spreaders can significantly impact the number of new infections. When such a singular person infects a large number of others, the tracing effort shifts to identifying who visited the same locations as the super-spreader. *Presence tracing* is related to proximity-based DCT but poses distinct challenges. As shown by the case of the Covid-19 outbreak in an LGBTQ nightclub in South Korea [261], the information on past visited locations can be highly sensitive. When faced with the risk of stigma and discrimina-

tion, people are increasingly motivated to circumvent the system. It is, therefore, important that privacy is preserved during presence tracing.

Up-to-date statistics on the situation are needed to be able to make sensible decisions on which measures are required to combat an epidemic or pandemic at a certain point in time. If data for such statistics is collected not only from diagnosed people but also from healthy people and those at risk, privacy becomes relevant. The problem of privacy-preserving data collection also exists outside of the pandemic context, for example, when sensitive data is collected for medical studies. Standard practices in crowdsourcing and data collection for medical data leave many attack vectors open for exploitation.

Providing efficient methods for privacy-preserving data analysis while ensuring technical privacy guarantees for data donors is challenging. Trusted Execution Environments (TEEs) provide a method for centralized data processing with provable guarantees. However, technical drawbacks of TEEs need to be addressed to ensure that the privacy of data providers is protected against both overly-curious data analysts as well as the infrastructure hosting the TEE, i.e. a cloud provider. Here, mitigating leaks through volume and access patterns from the TEE are the main focus.

1.2 OUTLINE & CONTRIBUTIONS

Being at the forefront of proposals for privacy-enhancing technologies for Covid-19 contact tracing, the author of this thesis contributed to both the research discourse as well as the public discussion [1]. This thesis bridges the gap between the context of privacy-preserving DCT and more general approaches to analyzing private data. All the while, we never lose sight of the question of how private data can be utilized without requiring trust to be placed in a central entity. In the case of DCT, this can be the government or a health authority. From a more general perspective, this entity can be the initiator of a crowdsourcing campaign or a data collector.

[1] *In collaboration with Samuel Brack, Jeanette Hofmann, and Björn Scheuermann, an article in April 2024 in the German online newspaper* `netzpolitik.org` *on different approaches to DCT [10] was published.*

This thesis is organized into three parts. The remainder of Part i presents preliminaries on privacy by design and adversarial modeling (see Chapter 2).

Part ii focuses on the subject of proximity-based DCT and presents multiple contributions in this area. To this end, Chapter 3 presents an introduction to the topic. By providing a taxonomy of contact tracing approaches based on Bluetooth Low Energy (BLE), we identify requirements regarding the functionality, security, and privacy of such applications. This allows the identification of common attack vectors and mitigation strategies. Solutions to these problems are relevant not only in contact tracing but also in other scenarios where data is collected from a crowd. We also reflect on the real-world impact of DCT.

The taxonomy of DCT application introduced in Chapter 3 is then used to organize the remainder of Part ii. Chapter 4 presents two approaches for client-side risk assessment. A main aspect of these two contributions is dealing with the challenge of establishing trust in the authenticity of warning messages while ensuring that no sensitive information is leaked to central entities such as the health authority or a messaging server.

Client-side risk assessment suffers from leaking timing information, which allows users at risk to identify the person who was the source of a warning message. Cryptographic protocols are powerful tools that can be used to provide privacy during contact tracing. They allow the computing of risk scoring without revealing the time of encounter with an infected person to users. These protocols can also utilize GPS location data instead of proximity data without revealing sensitive information. Chapter 5 presents and evaluates two such approaches.

As super-spreader events became a driving factor of the pandemic, more attention was placed on monitoring locations where people gather. Chapter 6 presents an approach to implement such a presence tracing system without centralized data storage on top of an already existing and widely established system for proximity-based DCT.

Gaining statistical insights into the pandemic based on DCT data proved to be challenging. Widely used proximity-based DCT systems relied on clients for risk detection as a measure to establish trust with users and mitigate surveillance. Centralized data stores without additional privacy protection go against the core idea of these DCT systems. As a result, relevant information is distributed between all users. However, knowledge can be gathered by crowdsourcing data from volunteers. This topic is discussed in Part iii. Common practices in this area, even for scientific studies, are far from privacy-preserving and leak sensitive information at various steps. In Chapter 7, we first identify common general practices and then analyze the resulting risks to privacy.

In Chapter 8, we present a privacy-preserving alternative for such data gathering and evaluation campaigns. The presented platform builds on TEEs for establishing trust and mitigating sources of private data leakage through oblivious algorithms and differential privacy.

2

ON PRIVACY BY DESIGN AND ADVERSARIAL MODELING

Protecting privacy is a tedious task, as sensitive data can emerge or resurface in various forms and places. Even data without identifiers can be used to draw meaningful information. As famously stated by a US general, "We kill people based on metadata" [171].

Software that follows the principle of *privacy by design* as well as *privacy enhancing technologies* aims to protect users and mitigate leakage of sensitive data inherently. Here, security serves as a pre-condition for privacy. Privacy can only be adequately protected if the proposed system works as expected and does not leave gaps for adversaries to exploit.

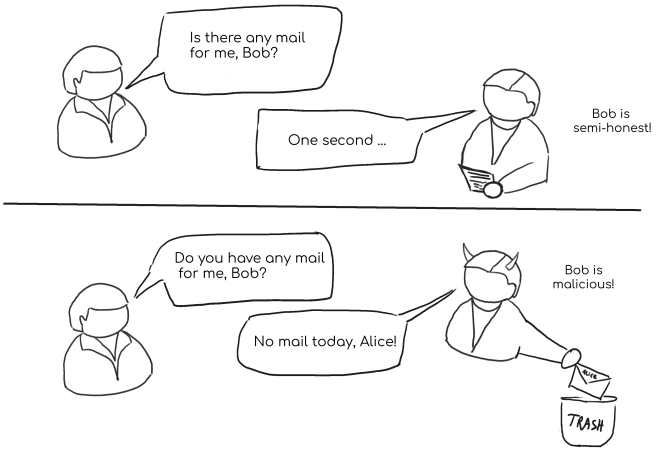


Figure 2.1: Comic explaining the difference between semi-honest and malicious adversaries.

When designing algorithms or systems that are private by design, it is essential to evaluate what type of attacker the users must be protected against. Defense mechanisms might differ when considering the attacker to be a well-equipped government, a company with commercial interests, or a tech-savvy malignant user. In research, the following adversarial models are commonly used to evaluate the threat against a system. A *semi-honest* or *honest-but-curious* attacker is interested in anything they can learn while following the protocol. In contrast, a *malicious* adversary will diverge from the protocol to learn secret information, as shown by the comic in Figure 2.1. Malicious behavior can include sending specially crafted messages, participating in a protocol with multiple entities in the form of a *Sybil attack*, or posing as a *man-in-*

the-middle. A *covert* adversary aims to appear honest but might behave maliciously if the risk of being detected is minimal.

An adversary might listen to traffic on the network or in the radio band to learn information. In this case, they are referred to as *network observer* or *eavesdropper*. This adversary can *passively* listen in or *actively* participate in the protocol. Note, that the thesis diverges here from the common notion in cryptography of what is considered a passive and active adversary.

In the following chapters, it is always important to keep these different types of attackers in mind. During Digital Contact Tracing (DCT), but also when computing statistics on sensitive data, it is crucial to always keep in mind where sensitive data accumulates and whether adversaries can gain access at some point in time.

Part II

THE PANDEMIC

3

AN INTRODUCTION TO PROXIMITY-BASED DIGITAL CONTACT TRACING

This section is based on a survey written in collaboration with Samuel Brack and Björn Scheuermann [2]. It was published in 2021 in ACM Health.

At the beginning of the year 2020, Covid-19 turned into a global pandemic challenging both healthcare systems as well as democratic institutions [60, 96, 147, 149]. To mitigate its spreading, social and economic life was shut down in affected areas [235]. Tools often used in the past for containing diseases had proven not to be effective enough to deal with this quickly spreading, highly infectious, and deadly virus [124, 284]. Therefore, new methods were developed to mitigate the pandemic such as the automation of contact tracing previously done manually by health authorities to speed up the process of discovering new infections. Early systems implemented by Singapore, South Korea, and Israel either used more data than necessary to fulfill the task, e.g., by collecting extensive data from healthy citizens or revealed private information to the public [147, 149, 289]. Concerns were raised about an increase in discrimination of socio-economic or ethnic groups through the adoption of automated or *Digital Contact Tracing* (DCT) [186]. In many countries, it was not feasible or acceptable for the state to enforce nationwide adoption of the local DCT application [19, 115, 195]. To ensure great effectiveness, it was therefore essential that citizens have sufficient trust into a DCT system to participate voluntarily. System designs that send detailed location or contact histories to a government-run central entity without any privacy protection might look more effective in the beginning. However, societies require transparent processes and data protection in exchange for their participation in the system.

Many privacy-preserving DCT systems were proposed and threats to privacy and security are manifold. In this introduction, we first take a look at the origins of DCT in Section 3.1. Then some terms and definitions on DCT are clarified in Section 3.2 which will be used throughout this first part of the thesis. Considerations regarding the sensors are discussed in Section 3.3. In Section 3.4, approaches for DCT are systematized which rely on Bluetooth Low Energy (BLE) for proximity detection. Approaches that protect the privacy of users are particularly emphasized. Based on this review, common challenges with regard to security, privacy, and functionality are established and solutions are discussed in Section 3.5 and Section 3.6. In the last part of this chapter in Section 3.7, social and societal aspects of DCT such as the adoption rates due to public perception, the usability of DCT apps, the overall effectiveness of DCT during the pandemic as well as its potential as dual-use technology are addressed.

3.1 A SHORT HISTORY OF DIGITAL CONTACT TRACING

Contact tracing is the process of finding new cases for a certain disease by retracing who had been in contact with a diagnosed patient. The standard practice is interviewing diagnosed individuals to manually derive potential contacts. This approach has been used in the past for various diseases like HIV, SARS, and Ebola [109, 315]. Both in theory and in practice it has proven to be a valuable tool for containing epidemics. Stochastic modeling was used to evaluate the efficiency of contact tracing, showing that the rate at which new infections are discovered cannot be considerably lower than the rate at which the infection spreads [109, 160, 315]. A direct requirement for contact tracing following this finding is that potential contacts are notified as fast as possible so they do not infect others. For this reason, most countries require persons infected with a notifiable disease to provide the responsible health authority with all relevant information for contact tracing [67, 121, 223]. This can be information about social contacts but also the history of recently visited locations. Standard or manual contact tracing is especially difficult for airborne diseases like SARS, MERS, or Covid-19 [109]. This is because contacts from random encounters cannot easily be notified as the diagnosed person can oftentimes not provide sufficient information.

Digital contact tracing, short DCT, aims to speed up the manual process. To ensure that warnings are delivered quickly to people who are at risk and to enable the notification of random encounters, it has become desirable to improve existing manual systems with modern technology [124]. Here, proximity data is used by the DCT app to inform users of past close encounters with people who were diagnosed. This enables fast testing and quarantine.

It has been discouraged to consider DCT a replacement for manual contact tracing [142, 258]. DCT systems might be faster, more scalable, and once installed less costly. However, the manual approach has been proven to be effective in epidemics before 2020, is already in place, and provides rich human-to-human interaction. Human contact tracers are also capable of detecting non-direct methods of transmission through questions. For this purpose, they require a diagnosed user's location history to trace potential contacts. Some DCT apps are specifically designed to support manual contact tracing processes [68, 101, 142, 266, 271]. DCT systems deployed to combat Covid-19 were generally used in combination with existing procedures.

Early research towards automated disease transmission tracking was done between 2007 and 2012 by the FluPhone project [127]. The goal of this project was to better understand and predict the influenza epidemic and how people alter their behavior in response. To this end, a field trial was conducted in which participants downloaded an app onto their phone that checked for other devices in proximity using Bluetooth [320]. For detecting phones close by, the FluPhone project built

upon Haggle [277], a design for ad-hoc networks using Bluetooth. Information about encounters of devices was sent to a central server using mobile data. GPS measurements were used to improve results. Participants reported symptoms using the app to determine if these indicated an influenza infection. The system also had the capability of marking devices as infected which could subsequently contaminate other users' devices they encountered based on probabilistic calculations.

Since then, research in the field of DCT has been slow but steady [24, 29, 125, 193, 238, 271, 273, 322, 323]. Besides the seasonal influenza [127, 323], diseases such as the equine influenza [29, 125, 193, 323], avian influenza [323], SARS [29, 193, 323], MERS [193], Ebola [29, 271, 273], and Zika [193] moved into focus. With the 2020 Covid-19 pandemic, many new approaches were proposed and implemented. The first country to roll out a full proximity-based DCT application for Covid-19 was Singapore with TraceTogether [287].

3.2 TERMS AND DEFINITIONS

To ensure a common understanding on DCT, a few terms need to be defined. Additionally, the potential attackers as well as their targets are listed. These definitions are relevant for this chapter and all following chapters of Part ii of this thesis.

3.2.1 *Relevant Terms*

To ensure common understanding, we introduce the following terms in the context of DCT.

1. *DCT system*: A DCT system consists of an app that can be installed on the users' mobile devices and a backend, typically a server. To function properly it is generally assumed that the local health authority operates the system.

2. *User*: Users of a DCT system are people who downloaded the app and have it activated.

3. *Diagnosed*: People are considered diagnosed if their infection has been medically verified and reported. DCT systems can only consider diagnosed people who have been using the respective system before they fell ill.

4. *Encounter*: When two users Alice and Bob are in proximity of one another, this is called an encounter.

5. *Contact*: If Alice is diagnosed as infected after an encounter with Bob, then Bob is called a contact of Alice.

6. *At Risk*: Users are considered at risk if they have had encounters with diagnosed people. This does not necessarily mean that they are disease carriers.

7. *Risk Scores*: Risk scores are calculated depending of the exposure of a user at risk. If the score exceeds a certain threshold, the user is notified.

8. *Pseudonym*: BLE-based approaches to DCT advertise ephemeral or static IDs. Such IDs are called a pseudonyms in this work.

3.2.2 *Definitions for Attacker Types*

When evaluating the security of any data processing system, it is essential to define the type of adversaries against which the system is secured. In DCT systems there are several parties with different prior knowledge and capabilities:

1. *Health authority*: This is the public institution tasked with containing the spread of the disease. It may have an interest in learning as much about users and diagnosed people as possible, for instance, their relations to each other or where they have been in the past. Another possible goal is the deanonymization of users at risk. Since infections with SARS-CoV-2, the virus causing Covid-19, have to be reported in many countries [67, 121, 223], it can be assumed that the health authority possesses a considerable amount of information about diagnosed users. In some legislation, it is even a crime to not support the health authority during contact tracing [223]. The health authority does not have an interest in blocking contact tracing or stopping risk notifications to users.

2. *Users*: Users want to determine their health status. They might also have an interest in figuring out who is infected or who infected them. It is important to distinguish the potential targets for this type of attacker. Attack vectors might differ between random users, close social contacts who are regularly in the presence of the attacker, or public figures who are easy to track down by a *curious stalker*. The stalker can for example follow victims and observe if their habits change.

3. *Diagnosed users*: Diagnosed people participate in most systems through having been reported to the health authority by their doctor. They have an interest in not revealing too much sensitive information about themselves to the public and the health authority because they fear public humiliation [261] or other forms of social punishment. But diagnosed users can also be malicious, by trying to figure out who they have infected or who was responsible for their infection.

4. *Eavesdroppers*: Eavesdroppers are passive attackers who listen to the communication of the protocol, both on the wireless network as well as the communication with a centralized backend. Users, networks, and service operators can all take the role of an eavesdropper in a protocol.

5. *Service operators*: The DCT service and its infrastructure can be run by the health authority or by a third party such as a contractor. Servers and cloud storage fall into this category. A service operator can try to learn general information about users and diagnosed people as well as their health status by observing and manipulating data passing their system.

6. *Network operators*: Network operators can have similar goals as service operators, but are only capable of observing and manipulating data that is sent through the network.

3.3 SENSORS FOR PROXIMITY DETECTION

An important part of DCT is determining if users get into close contact for at least the duration that makes disease transmission possible. Smartphones have been the main focus for DCT systems since they are widespread and provide a variety of sensors that can be used for proximity detection. In the following, smartphone sensors are presented and discussed which allow proximity detection or distance measurements between users.

3.3.1 *Bluetooth*

During the last few years, *Bluetooth* has emerged as a useful technology for measuring the proximity between devices. Bluetooth can be used for positioning and proximity detection, especially in indoor settings. By using the Received Signal Strength Indicator (RSSI) to estimate the distance between a receiver and a transmitter, relative or absolute location information can be derived. Raghavan et al. [256] were able to show that Bluetooth version 2.0 can be used for localization with an error of less than 45 cm. Liu et al. [210] demonstrated that Bluetooth is efficient for detecting face-to-face interactions by providing a model for estimating distance using RSSI readings. Bluetooth has the problem that it is an active protocol where a connection must be established between the two parties before any payload can be exchanged. This potentially hinders an effective exchange of messages due to the added complexity of the connection establishment. Additionally, since devices advertise themselves, they signal to possible attackers where to find an activated interface. This can then be exploited to hack the device using known vulnerabilities of the protocol and its implementations [226].

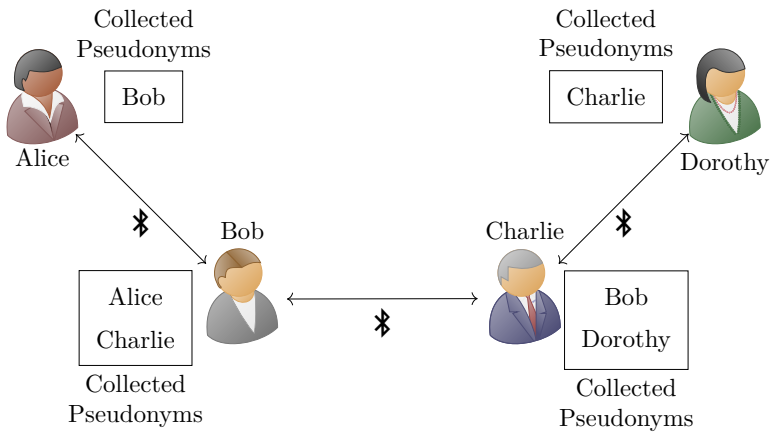


Figure 3.1: During contact collection, each user stores the IDs of all devices that are in proximity. These IDs can be used to notify close contacts in case of a subsequently detected infection. Figure from [2].

3.3.2 *Bluetooth Low Energy*

Bluetooth specification 4.0 introduced *Bluetooth Low Energy* (BLE), an energy-efficient, short-range variant of classic Bluetooth [105]. Classic Bluetooth and BLE are not interoperable. In 2020, both Bluetooth and BLE have a high adoption rate, as 100% of new smartphones support both standards [53]. Due to its battery-saving properties, BLE was adapted for positioning and proximity detection [117, 118, 230, 262] and is especially interesting for mobile use cases. Bertuletti et al. [51] were able to reduce the error of BLE-based location measurements to less than 40 cm. Similarly to distance measurements for classic Bluetooth, the RSSI is used to determine the distance between sender and receiver. Different models for signal propagation can be employed for this purpose, such as exponential or polynomial approximations.

To transmit data over BLE, a device sends broadcast packets during each of its *advertisement intervals* to the three available channels. Recipients use the *scanning mode* to listen for such advertisement packets [54]. During each scanning window, they record transmissions. Scanning can be conducted either actively or passively. The active scanning allows to request additional data from the advertiser. When scanning passively, devices do not establish a connection between each other. Instead, scanning devices simply extract information from broadcast messages. The original use case of BLE broadcast messages is periodic sensor readings, but each BLE-capable device can be configured to advertise short data packets as well. A device cannot scan and advertise at the same time. The durations for both advertising and scanning windows are configured locally for the device. Timings of these scans have to be considered so that each device has reasonable chances to see all others and can also be seen [207].

Passive scanning, in combination with its energy-saving properties, makes BLE better suitable than Bluetooth for proximity detection and distance measurements. BLE was therefore widely adopted as the technology for realizing DCT. In BLE-based DCT, users continuously transmit pseudonyms via broadcast messages to everyone nearby (see Figure 3.1). These messages can be received and recorded by other users. If a person is diagnosed, the pseudonyms they have seen in the past (as well as the distance to them) are used to identify their random encounters.

One shortcoming of BLE for proximity detection and contact tracing is the large variability of transmission power across different smartphone types. RSSI readings have to be calibrated to the respective devices [142]. Additionally, distance measurements can be noisy due to multi-path and shadowing effects. These are caused by objects such as walls or furniture absorbing or reflecting radio waves. The authors of DP-3T [305] noted that such errors generally increase the measured distance and rarely decrease it. So instead of solving the problem of correct distance estimation, they focused on determining if the distance is larger than a certain threshold. For this purpose, they conducted experiments for various everyday settings [13]. With a precision of 80% and a recall of 52.5%, they were able to identify if the distance was larger than 2 m for all scenarios. Sattler et al. [275] were able to correctly identify 100% of risky contacts with a duration of 15 min at 2 m distance while accepting a false positive rate of 30%. Experiments conducted with soldiers of the German army have also shown that a mapping from RSSI to infection risk can be reasonably accurate [215]. However, BLE-based proximity detection is not effective and unreliable in public transit such as light rail trains due to large measurement errors [198]. Additionally, there is a disparity in detection rates between different phone operating systems as Android phones seem to have an overall better proximity detection rate than iOS phones [111].

For usability reasons, it is essential that a DCT application can run in the background. Apple's iOS restricts the usage of the corresponding interfaces for apps running in the background, thereby interfering with proximity detection [142]. This limitation on iOS concerns all types of sensors except those related to location tracking.

The simplicity of BLE does not come without limitations. The payload in an advertisement packet is limited to 31 bytes [54]. In the energy-saving passive scanning mode, active approaches for exchanging data cannot rely on the advantage of protocol confirmation messages. Approaches that rely on multiple packets being exchanged between users depend on the usage of the active scanning mode and both devices being visible to each other for some time, a requirement that can be difficult in mobile or crowded scenarios.

Both, sending advertisement packets and scanning, require energy. Thus, a tradeoff has to be made between saving energy and being ac-

tive on the BLE band to participate in DCT and interact with all other devices. Especially for mobile devices this tradeoff is an important part of the system design. Approaches that work with fewer interactions (i. e., passive scanning only) can save considerable amounts of energy compared to systems with active message exchanges.

Similarly to Bluetooth, BLE also has vulnerabilities that make it exploitable to attackers when turned on. These are for example SweynTooth [131], CVE-2019-2102 [227] as well as device fingerprinting based on imperfections in BLE components [133].

3.3.3 *GPS, Cell Tower Triangulation, and other Methods*

Bluetooth and BLE are not the only technologies available for determining proximity or co-location. Methods like GPS [220, 5], cell tower triangulation [147], Wifi [29], or correlating magnetometer readings [169, 238] can also be used for DCT. However, all of these technologies have shortcomings which we will discuss in the following. *GPS data* is generally seen as very privacy sensitive, as it can reveal identifying information about a person like their home and work address. At the same time, its resolution is not fine-grained enough to detect face-to-face interactions between people, especially in areas with tall buildings or indoors [179]. Covid-19 is an airborne disease, so while being in the same room as an infected person without protection is dangerous, sitting on the other side of a wall is not. These kinds of false positive errors are difficult to mitigate when using GPS or cell tower triangulation. Both technologies are too imprecise to derive meaningful data about the interactions of users.

Wifi, just like Bluetooth/BLE, has the advantage of being blocked by objects such as walls. This ensures that people do not log an encounter if they are separated by a barrier strong enough to mitigate the spread of virus-containing aerosols. While Wifi has been widely used for indoor positioning [244], just like cell tower triangulation it requires infrastructure that might not be available everywhere, especially outdoors or in remote locations. It is therefore not suitable for DCT, which is required to function anywhere.

Correlating *magnetometer readings* of users is another passive method suitable for DCT. It requires little energy while working indoors and outdoors. When two magnetometer readings have a similar variance during the same time period this indicates that they were recorded at the same location. No information about the distance between the people recording these traces can be deduced. However, proximity information is crucial for evaluating the likelihood of transmissions in a DCT setting [296]. There has been little research in the area of proximity or co-location detection using magnetometers so far and it is not as well investigated as BLE. So while this method works in the laboratory, reproducing the findings on a large scale might be difficult, making

this technology inadequate for DCT during the Covid-19 pandemic as timely deployment was vital.

The Fluphone project also tested *Radio-Frequency Identification* (RFID) tags [127] to detect proximity. While this approach is interesting, tags need to be distributed to all users. This overhead is considerably larger than providing an app in various app stores and using common smartphone capabilities. Since the ID of RFID tags is static, this technology also allows the re-identification of users, making it easy to track their location.

Another tag-based approach was tested in a Singaporean hospital [159]. Tags for a *Real-Time Location System* (RTLS) were handed to patients and staff. Additionally, corresponding transmitters were placed in the building. While the rate of detecting contacts was high, the setup costs impact the applicability of this technology to larger areas and privacy is not protected.

Table 3.1 summarizes all mentioned sensor types and their different properties with regard to proximity and co-locations detection.

As we have seen, BLE is the most suitable base technology for DCT. For this reason, most systems proposed and deployed are built on this approach to detect contacts between users. The main differences between various approaches to DCT lie in the way how risk assessment is conducted and which parties hold relevant data. In the remainder of this chapter, we will therefore focus on works using BLE for proximity detection.

Table 3.1: Properties of different sensors available for DCT. ⋆ Requires active communication. † More energy efficient than Bluetooth. Table as in [2].

Sensor	Precision Sufficient	Distance Measurement Possible	Privacy(P)/ Security(S) Issues	Infra-structure Required	Runs in Background (iOS/Android)
Bluetooth ⋆	✓	✓	S		x/✓
BLE †	✓	✓	S		x/✓
GPS		✓	P		✓/✓
Cell Tower Triangulation		✓		✓	✓/✓
Wifi	✓			✓	x/✓
Magnetometer	✓				x/✓
RFID	✓	✓	P	✓	x/✓
RTLS	✓	✓	P	✓	-

3.4 SYSTEMATIZATION

Systems for DCT can be categorized by whether the risk assessment is done on the server or the client side. However, for both concepts, subgroups can be distinguished with similar shortcomings and advan-

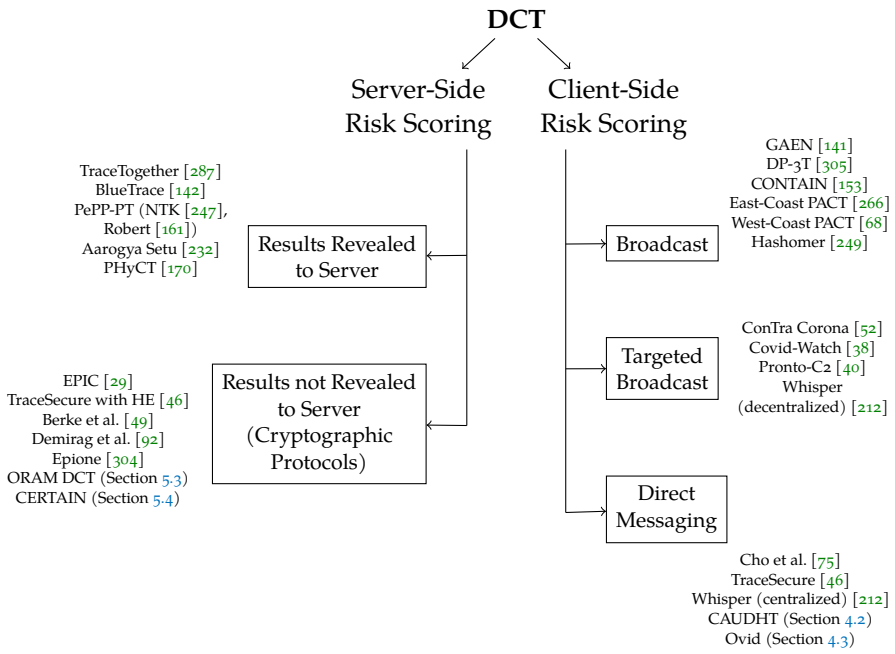


Figure 3.2: Systematization of DCT approaches.

tages. See Figure 3.2 for an overview of the systematization and the categorized DCT approaches.

3.4.1 *Server-Side Risk Assessment*

The most straightforward approach to server-side risk assessment is relying on the server to manage risk assessment. However, more private solutions that leverage cryptographic privities also fall into this category. Approaches to server-based DCT can be distinguished whether the results of risk scoring are *revealed to the server or not*. See Table A.1 in Appendix A for an overview of all the approaches mentioned in this section as well as their privacy and security properties.

Results Revealed to Server

A simple approach to risk assessment is computing risk scores based on proximity data in clear on the server side. This means that results are revealed to the server and its operator, as shown in Figure 3.3. Users can either upload the pseudonyms that they used in the past when conducting risk scoring, have their pseudonyms assigned to them by the server, or upload their own recorded encounters. Depending on the variant, the diagnosed users must provide the corresponding data. The first variant comes the closest to the privacy trade-off of manual contact tracing. Some examples of approaches that reveal the result of risk assessment to the server are TraceTogether [287], the corresponding

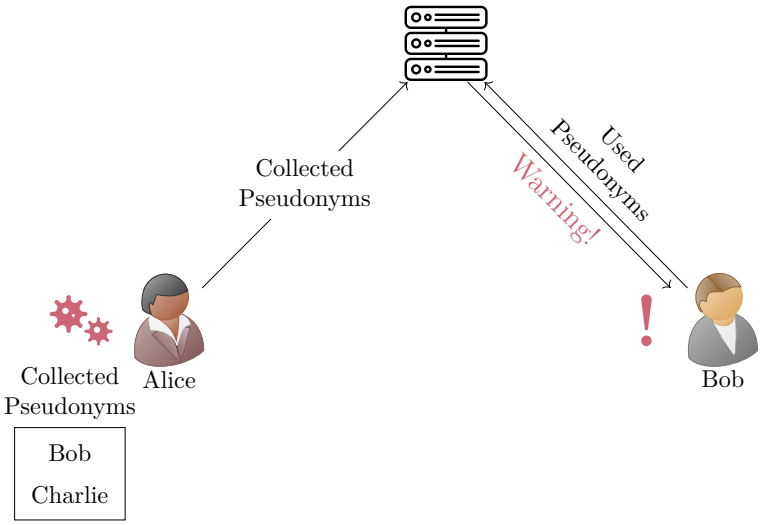


Figure 3.3: An exemplary server-based DCT model where the server learns the risk scores. Alice sends her collected pseudonyms to the server when diagnosed as infected. The server does a risk assessment for her contacts and warns Bob. Figure adapted from [2].

open-source project BlueTrace [142], Arogya Setu [232], PePP-PT [247] and PHyCT [170].

While using a central server provides certain advantages, it also has implications for the security and privacy of the users. The identities of people who should quarantine are revealed to the health authority, and restrictions on these people can, therefore, be enforced. No data is revealed to users other than the risk notification received by users at risk. Recipients can only guess that they might have been infected by someone from their history of encounters. But since proximity measurements are made independently, both sides of an encounter might record different distances and an encounter might have only been recorded by one side. A malicious user cannot rely on using their history of encounters when trying to figure out who caused a risk notification. This means this type of approach protects the identity of diagnosed individuals against other users.

Instead, the dangers of systems revealing risk assessment results to the server lie elsewhere. Information about the contacts of users is leaked to the entity running the servers, which is either the health authority or a service operator. In case a user is reported as a contact by several diagnosed patients, the server can directly derive that these people might know each other. It also learns about relations between healthy users as the server can observe that some users always appear at the same time in collected data sets. Using additional information such as the time of an encounter or other prior knowledge, specific details about the nature of users' relations can be revealed. While these

individual relationships might seem insignificant, this attack vector allows the adversary to build a social graph for parts of the user base.

For systems where the health authority always knows who uses which pseudonyms, a malicious health authority could install Bluetooth sensors in popular areas like train stations and collect pseudonyms there. This allows the health authority to learn the location history of any user who passes the capture device. Depending on how tightly knit the infrastructure of publicly located Bluetooth sensors is, the health authority can follow every movement of users.

Another issue arises from the way ephemeral pseudonyms are linked to static ones at the backend. For example, in PePP-PT [247], ephemeral pseudonyms are created by encrypting a static identifier. The reference implementation of Bluetrace [142] works similarly. In this case, if the encryption key is leaked, all identifiers issued with this key become linkable and recorded BLE traces can be anonymized by any eavesdropper on the BLE band. Rotating keys have been proposed to reduce this threat [311]. An attacker observing the network does not learn who is at risk, but uploads to the server will reveal who is diagnosed if no additional measures such as cover traffic or hiding the IP address are taken.

As explained in the motivation of this chapter, it is essential that users trust the contact tracing system enough to participate voluntarily. Many people seem to be deterred by systems they find too intrusive or incapacitating, such as one where they are forced into quarantine instead of taking the decision themselves [37]. As the approaches discussed in this section do not provide privacy guarantees, misbehavior of the risk assessment server can not be detected. This allows them to be misused for crowd control. There is also the fear that the described server-based approaches facilitate the creation of new surveillance infrastructure that could, for example, be used to target minorities [37, 60, 64]. These two aspects have greatly influenced the public discussion in some European countries causing governments to move away from approaches like the ones described in this section [85].

Results not Revealed to Server

Using cryptographic protocols such as *homomorphic encryption* or *Multi-Party Computation* (MPC) can mitigate certain attacks on the privacy of users while providing similar functionalities as non-private server-based approaches. Especially *Private Set Intersection* (PSI) is the core functionality of many such protocols. See Figure 3.4 for an exemplary approach using cryptographic protocols for risk assessment. Some examples for such DCT approaches are EPIC [29], the homomorphic encryption variant of TraceSecure [46], the proposal of Berke et al. [49], the protocol by Demirag et al.[92] and Epione [304] [2]. In Section 5 of

[2] *Section 5.1 will discuss these approaches in detail.*

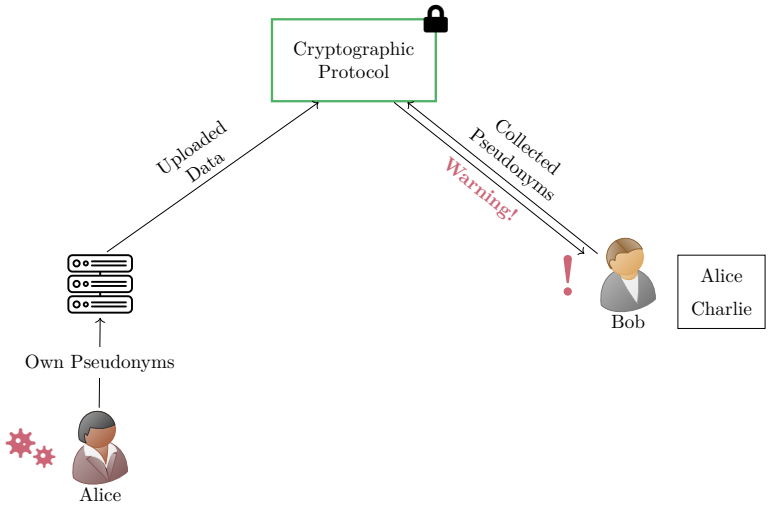


Figure 3.4: This figure illustrates how a simple PSI protocol for BLE-based DCT could work. This example does not leak the intersection to Bob. Figure adapted from [2].

this thesis, two contributions to DCT are presented that rely on *Oblivious Random Access Memory* (ORAM) and PSI primitives.

DCT systems which use cryptographic protocols are cryptographically secure, meaning they leak no more information than intended. Approaches using MPC, for example, allow to hide the data provided by the user from the central entity. Additionally, all MPC protocols can be secured against malicious attacks by accepting performance penalties [134].

Homomorphic encryption and MPC protocols have a significant computation and communication overhead which can cause a long protocol runtime. Distributed Denial-of-Service (DDos) attacks against the resources of a central computation server are, therefore, very effective. Due to the complex operations executed by the central server, an attacker could aim to exhaust the server's resources by sending randomly generated data.

Another downside of MPC protocols is that they often require many gigabytes of data to be communicated between different parties. This is hardly feasible on metered mobile data connections. Mobile energy consumption is also limited due to battery sizes. More importantly, even the general public acceptance of DCT relies on its usability on mobile devices. This problem can be partially eased by securely moving the computation load from end devices to the cloud [185].

While the protocols themselves may be secure, it is difficult to ensure that inputs are not just a subset of the recorded pseudonyms. A malicious user trying to find out if a target Tiffany has recently been diagnosed can alter their input data to only contain Tiffany's pseudonyms. If a positive risk score is returned, the attacker learns that Tiffany's pseudonyms are contained in the health authority's data set. She, therefore,

must have been recently diagnosed. This type of attack also works for a malicious health authority. A malicious health authority can alter its input and only use a subset of its data or add specifically crafted data. This way, it can make sure that an infection risk is detected by specific users, causing them to believe that they are at risk even if this is not the case.

3.4.2 *Client-Side Risk Assessment*

A different type of DCT is based on the idea that the risk status of a user should be calculated locally on the client's device and not be revealed to the health authority, service providers, or network providers. Data passes these infrastructures, but no information about social interactions is revealed and either diagnosed users and/or the users at risk remain private. This technique often requires more resources on end devices than the straightforward server-based approach. Several models using client-side risk assessment are discussed in this section. We distinguish between systems using *broadcast*, *targeted broadcast*, and *direct messaging*. A broad overview is provided to distill common properties for each category.

Details for the individual designs mentioned in the following are presented in the related work section of Chapter 4. In Appendix A, two tables compare the individual DCT designs and their properties. Table A.2 focuses on broadcast and targeted broadcast based designs while Table A.3 provides an overview of direct messaging approaches.

3.4.3 *Broadcast Models*

Broadcast models rely on distributing information about diagnosed individuals to all users who use this information to conduct local risk assessment. As long as users are undiagnosed, they generate pseudonyms on their end devices for advertisement and record the pseudonyms of others close by. In case a user is diagnosed, they upload all their past pseudonyms (or the key material required for generating these) to the server. Daily, all users download the recently published pseudonyms (or the corresponding key material). They then check locally if a contact with a diagnosed person has been recorded. This type of DCT is often also called *decentralized*.

Apple and Google, two companies that together dominate the market for smartphone operating systems, formed an alliance in 2020 to present a joint approach for DCT [141]. This thesis will refer to this proposal as *Google Apple Exposure Notification* (GAEN). They proposed and implemented a technical specification for an API that was provided by their smartphone operating systems. To derive the next pseudonyms, a day-specific key and the epoch number identifying the current time interval are used as input for a pseudo-random function. When a user is diag-

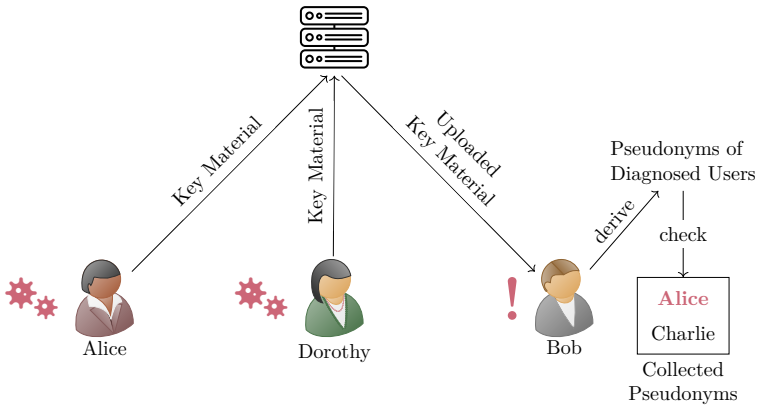


Figure 3.5: Broadcast-based DCT as performed by GAEN. When Alice is diagnosed, she will upload the key material for the pseudonyms that were advertised when she was infectious. Bob will download a list from the server containing Alice's data, as well as data of other diagnosed users such as Dorothy. Checking locally against his list of recorded pseudonyms, he recognizes a past encounter with Alice.

nosed, they upload the daily keys for the relevant time period. Other users then use this to locally derive the corresponding pseudonyms and check for encounters. Figure 3.5 illustrates the notification process of GAEN. The implementation of the DCT app and setting up server infrastructure are tasks left to the health authorities that are interested in using GAEN. GAEN was available for Android 6.0 and higher until September 2023 [79, 136]. On Apple devices, it was introduced with iOS 13.5 [34]. Many DCT applications deployed in real-world settings during the pandemic relied on the broadcast model as these apps were built on GAEN API [141]. This includes the German Corona-Warn App, the Swiss SwissCovid App, the Italian Immuni App, and at least 62 others [95, 257]. Without using the GAEN interface, reliably sending and scanning for BLE advertisements in the background was and is still not feasible [237]. This resulted in multiple governments abandoning projects that did not use GAEN [303]. The unwillingness of Google and Apple to provide a more low-level programming interface to the BLE stack has been heavily criticized, especially by France [303]. It has been argued that by making other types of DCT apps impossible, the companies undermine the sovereignty of governments.

Designs similar to GAEN are the DP-3T low-cost design [305], CONTAIN [153], East-Coast PACT by Rivest et al. [266], and West-Coast PACT by Chan et al. [68]. Differences between these schemes and the GAEN framework are mainly on an implementation level. Broadcast-based approaches with slightly different properties are the DP-3T unlinkable design [305] and Hashomer [249].

Approaches using the broadcast model can hide from the health authority the fact that someone has been in contact with a person who has tested positive. This can be an essential feature to gain users' trust, as they can review warnings for plausibility and are free to decide for themselves when it is time to seek medical attention. Since the risk status is calculated locally and all users receive the same data, service providers and network providers cannot guess a person's health status by eavesdropping. Broadcast models have the common weakness of revealing the pseudonym and approximate time when the encounter occurred to the user at risk. Overly curious users could try to abuse this information to deanoymize diagnosed people. This also simplifies attacks where a security camera is combined with a Bluetooth sensor device. Here, the captured data allows the attacker to connect the pseudonyms of diagnosed users to faces.

Another issue is impersonation attacks. A diagnosed user could upload different pseudonyms than the ones they used themselves to make it seem like someone else is infected. This class of attacks requires the attacker to gain access to recent pseudonyms of a victim, which can be obtained by sustaining physical proximity to the targeted victim. In some cases, access to the keys that are used for pseudonym generation is required. This can only be done by breaking into the victim's phone. A successful untargeted impersonation attack would require the attacker to guess a valid pseudonym. This is very unlikely to happen due to the high entropy of randomly generated pseudonyms.

Since risk scoring is done locally in broadcast-based DCT, a network operator will not know who is at risk. Nevertheless, through uploads to the server it is possible for the network and the service operator to learn who has been diagnosed.

3.4.4 *Targeted Broadcast*

In targeted broadcast systems, only the user at risk (and sometimes the corresponding diagnosed person) is capable of identifying that a broadcasted message was directed at them. As a result, people who did not get close to someone later diagnosed as infected or those for whom the exposure duration was not long enough will not receive a warning. This means that users cannot learn that someone they have seen in passing was diagnosed. Compared to the simple broadcast approaches presented above, a pre-filtering of users that need to be warned can be applied by the diagnosed individual. Examples of DCT systems that fall into this category are ConTra Corona [52], Covid-Watch [38], Pronto-C2 [40], and Whisper [212].

Many risks of targeted broadcast approaches are similar to those presented in Section 3.4.3. For all the analyzed systems relying on the idea of targeted broadcasting, it is possible for a user who stores a list of received advertisements to find out who was responsible for a

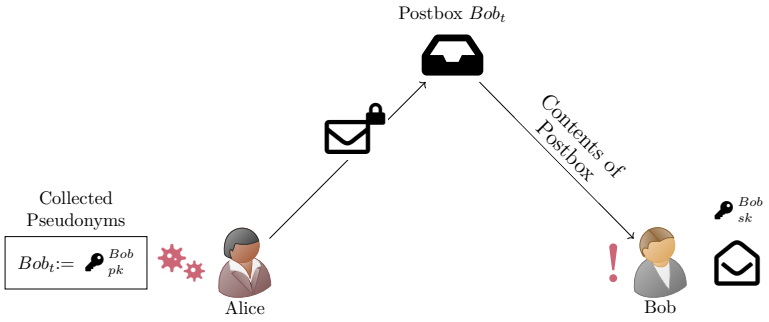


Figure 3.6: An example of a direct messaging approach to DCT following Cho et al. [75]. Alice collected Bob's pseudonym Bob_t earlier. She uses it to encrypt a message for Bob. This message is placed in the corresponding postbox where Bob can retrieve it. After its decryption, he knows that he is at risk. Figure as in [2].

received warning and is therefore infected. This can be improved so only the epoch of the encounter is leaked by breaking the link between the identity used for warning and the one used for advertising [52]. However, this still gives the user at risk the possibility to greatly reduce the pool of people that could have possibly caused the warning.

As the messages broadcasted are only meaningful to the designated receiver, no one else will learn if they are at risk. This is the case even if both sides of an encounter have measured different distances to one-another and the user at risk did not register the encounter.

3.4.5 *Direct Messaging*

Another way of doing client-side risk assessment in the context of contact tracing is direct messaging. Here, diagnosed users send warnings to those they have been in contact with. For addressing, either the pseudonym or information derived from the pseudonym is used. The warnings are relayed to the receiver by a server. Examples for this approach include the proposal of Cho et al. [75] and TraceSecure [46], as well as CAUDHT and Ovid (see Chapter 4). See Figure 3.6 for a simplified graphical explanation based on the proposal of Cho et al. [75]. Systems that might appear to leverage strategies of targeted broadcast but allow users to download data that is only intended for them or otherwise leak the graph of social interactions between diagnosed and undiagnosed users also fall in the category of direct messaging. This is the case for a centralized variant of Whisper [212] and a variant of ConTra Corona [52].

Some direct-messaging designs rely on a partially trusted central party. This is due to the fact that the central server can learn the underlying social graph or even parts of the identities of diagnosed users. Decentralizing this central infrastructure by distributing it over well-known independent entities can mitigate this issue. Cover traffic is

another option to hide the social graph. This has the advantage of additionally protecting against network observers trying to deduce users' health status from traffic patterns. However, allowing arbitrary traffic makes mitigating spam difficult. Attackers can try to congest a specific postbox so that the corresponding user will not be able to receive valid messages for their pseudonyms. Another issue is the aspect of authenticity. Malicious users might try to cause panic by sending "I am diagnosed" messages to many people without actually being at risk.

In Chapter 4 of this thesis, two contributions to the field of direct-messaging DCT systems are presented that provide solutions for the problem of authenticity and cover traffic.

3.5 FUNCTIONALITY ASPECTS

Aside from determining the risk of an individual user, there are various functionalities that a DCT system needs to provide. Some features enhance the capability of a DCT system by fulfilling additional needs of the users or the health authority. In the following, we will discuss functionality issues as well as additional features and how they were integrated by DCT systems. Table A.4 in the appendix provides an overview of the topics discussed in this section.

3.5.1 *False Positives and False Negatives*

An issue often mentioned when discussing the applicability of DCT are false positives and false negatives. A false positive in the case of DCT can belong to one of two categories. One option is that the situation for an encounter did not occur at all but a warning is still received. In the other case, the DCT system detected an encounter even though the transmission of the disease is highly unlikely, e. g., when two users were separated by a wall. Reasons for such errors can be manifold. To minimize the number of false positives based on distance, one option is to lower transmission power or improve the model for distance estimation, e. g., by having the sender provide information about its current transmission power or by calibrating the sender [142]. To ensure that an encounter is relevant, only those with a significant time span must be taken into account. For example, field trials were conducted to investigate the detection of an encounter for different phone positions and distributions of people [247]. Instead of detecting distance, some projects instead focused on detecting if the distance is more or less than 2m instead of measuring the real distance [15].

Reducing the number of false positives can lead to an increase in false negatives. As a result, risky encounters might not be detected and users who are at risk might not warned by the system. Here, the solution would be to increase transmission power while ensuring that other measures are in place to mitigate false positives, such as only

sending a warning if both sides have registered the encounter. The balance between both types of errors is important [161].

3.5.2 *Pseudonym Rotation*

How often pseudonyms are switched is a key factor in all BLE-based DCT systems. To mitigate tracking, pseudonym advertising epochs are generally kept short. Nevertheless, switching pseudonyms too frequently can become an issue when uploading data in case of an infection. The duration can vary from five minutes to 15 minutes up to 45 minutes in extreme cases where a long duration is a technical requirement [52]. A small value for the rotation period does make location tracking harder. At the same time, depending on the system design a small value can make it more challenging to recognize relevant encounters.

When rotating pseudonyms, it is essential to keep in mind that the device address of the BLE sender also changes regularly. To ensure privacy, the rotation period of the ephemeral pseudonym should not be smaller than the rotation period of the device address because otherwise, the load on the DCT system is increased without improving privacy [161]. In case longer durations are required, only multiples of the rotation period of the device address can be used safely.

As of Bluetooth Version v5.1, devices keep their address for 15 minutes. The GAEN initiative by Google and Apple had the advantage that its developers did not have to take existing values for device address rotation into consideration due to their access to the code of the internal Bluetooth stack. For this reason, GAEN [140] uses a period of 10 minutes.

3.5.3 *Authenticating Uploads*

DCT systems require an interface to both the testing infrastructure and the users to distribute meaningful risk notifications. Ensuring that a user is diagnosed while at the same time providing privacy is important. If there is no control over who is capable of uploading data to the system, trolling and planting fake data become possible. This makes warnings issued by the DCT system unreliable. The simplest solution to this problem relies on healthcare providers to collect data from diagnosed individuals which they then upload to a server. This way the system can be sure that as long as the healthcare providers are authenticated and trustworthy, the uploaded data is authentic. A more widespread approach is the use of token-based systems that allow diagnosed users to upload their data to a server after having received confirmation from a doctor. Tokens can either be handed out when the infection is verified or at the time of testing if an additional activation mechanism is used. The most simple approach to this is a verification code as used by GAEN [137]. It is also possible for users to commit data when being

tested which will be uploaded later [18]. If they test positive, the health authority authorizes the upload. This mechanism ensures that the data is not tampered with after the infection is verified. It also stops users from giving their upload tokens to others.

It is also possible to use blind signatures to prove to users that a warning or upload originates from a diagnosed individual (this will be discussed in Section 4.2) [40, 1]. This means the user can be certain about the authenticity of a warning. In this situation, a hacked malicious server without access to the health authority's private key can only delete warnings but not insert new ones. For the health authority to know who to issue blind signatures for, another token-based access mechanism needs to be used.

Another option to solve the issue of authenticating uploads is creating an infection certificate by having diagnosed users generate a new public key pair that will be signed by the health authority [153]. This certificate can be provided on upload to verify the infection. To ensure privacy, this mechanism relies on the non-collusion of health authority and upload server.

In broadcast-based DCT systems, a diagnosed user will upload their own pseudonyms to warn others. To ensure that a diagnosed person or a person in possession of an upload token does not upload the pseudonyms of someone else, a key derivation schema can be used [305]. Pseudonyms are not independent but derived from the same key material. To notify others that they have been diagnosed, users present the key material, which allows others to derive all pseudonyms belonging to this person. While this is useful for ensuring that no bad data is sneaked into an upload, attacks using the time slots of received warnings to deanonymize diagnosed people become more effective if all pseudonyms of a person can be linked.

3.5.4 *Verifying Encounters*

Imagine a black market where people offer money for faking encounters of a target person Tiffany with infected people. A freshly diagnosed user can alter the data they upload so an encounter with Tiffany appears in their encounter history. The attack can be stopped by having Tiffany check if she has recorded a corresponding event. This is done by design by broadcast-based DCT systems where only the advertised pseudonyms of diagnosed individuals are published [68, 141, 153, 249, 266, 305]. Some, but not all, systems relying on client-side assessment have similar checks [40, 1, 212].

Server-based approaches do not defend against this attack since the server does risk scoring. Here, a malicious diagnosed user can upload pseudonyms of the target Tiffany. The server will recognize the pseudonyms and send a notification to Tiffany. She will not be able to tell the

difference between a misbehaving server, fake data inserted by other users, and a real warning.

Liu et al. [209] propose a solution to the problem of verifying if a close contact occurred. When users have an encounter of meaningful duration, they initiate an active exchange over Bluetooth to swap identifiers and signatures. Later, zero-knowledge proofs [3] are used to demonstrate to the health authority that an encounter occurred. Many designs avoid the use of active protocols as this can make it easier for an attacker to exploit the device (see Section 3.3) [40].

[3] *Zero-knowledge proofs allow a prover to cryptographically convince a verifier of the properties of some input data without revealing that data* [291].

Another option to verify encounters is using cryptographic hashes [304]. Users compute a daily hash for their history of contacts which they upload to the server. To verify the hash, a zero-knowledge proof has to be used. This mechanism ensures that an attacker cannot forge queries or only use a subset of the data.

3.5.5 *Incomplete Reports for Privacy*

For privacy reasons, users may want to have control over what data they report in case of diagnoses. One option is to turn off BLE when they do not want any data to be collected or advertised, e.g., when visiting a sensitive location. Whether or not a diagnosed user provides their data for DCT is mostly voluntary. Some DCT approaches [40, 249, 266] additionally consider the option for users to only upload some data to the server. While being a privacy feature, this leaves room for extortion. In cases where risk scoring is not done locally but on the server side without cryptographic protection, a diagnosed person could blackmail others by threatening them to include their data in their upload to the health authority. Such a threat would only be effective if measures against users at risk were enforced.

3.5.6 *Proving Risk*

During the beginning of the pandemic, getting tested for Covid-19 was difficult as PCR tests were rare and the faster antigen test had not been developed yet [119]. It was, therefore, suggested that users who have received a warning by a DCT should have a right to be tested. For systems that employ server-side risk assessment and reveal the results to the server, it is easy to determine who is eligible for a test and the servers provide some degree of validation.

For systems where no central entity is informed about results from risk assessment, the process is more complicated. Even if a user receives a notification, they have to prove they are not simply forging encounters and notifications to get tested. For systems that rely on asymmetric key cryptography, the possession of a private key corresponding to an at-risk public key can be used as proof.

To prove exposure, by using a specifically designed key derivation scheme a verification key can be derived from parts of the advertised pseudonyms [249]. When a user becomes diagnosed, they upload this key to the health authority which is then distributed to all users. Users who want to prove they are at risk can present the corresponding collected pseudonym. Using the verification key, the health authority can figure out if the collected pseudonym belongs to a diagnosed person. This approach opens up new ways for the health authority to derive relations between users and does not prevent the transfer of pseudonyms belonging to diagnosed people to other users.

Another proposed option is to incorporate a random value u into all pseudonyms that can later be presented in a non-interactive zero-knowledge proof to the health authority to verify ownership [52]. To discourage people from giving away their proof, u can include a timestamp and the user's real identity.

3.5.7 *Statistical Insight into the Pandemic*

Statistics are a useful tool for health authorities and governments to take informed steps to mitigate the spread of a disease. The notification chain of newly diagnosed cases allows health authorities to stay informed. Regardless, delays in the notification processes and an overloaded health authority can cause these statistics to be less reliable. The motivation behind some server-based contact tracing approaches was to gain insights into the pandemic in real-time [161]. By monitoring the number of recorded beacons per person as well as the number of warnings sent during a certain time period it is possible to extrapolate the day-to-day number of infections. Such statistics also allow for determining the effectiveness of a DCT system by linking the number of warnings to the number of new cases.

Approaches where only the user learns their risk instead of a centralized entity can suffer from a lack of statistics. However, it is possible to compute statistics on distributed data. Google and Apple provided an extension of their GAEN protocol which uses the Prio protocol [176, 254] to compute differentially private aggregate statistics which were made available to participating health authorities [139].

Another option for collecting statistics in a privacy-preserving way by using trusted execution environments will be discussed in Chapter 8.

3.5.8 *Dealing with International Travel*

To facilitate cooperation between different states, systems were proposed for the federation between different health authorities [161, 247]. For instance, a country code is added to the pseudonym when it is transmitted. DP-3T [14] pays particular attention to the aspect of interoperability across borders, allowing users to enter regions they will

travel to or have returned from. When diagnosed as infected, users upload their history. If they indicate travel, the backend will communicate the pseudonyms of the user to the backend responsible for that specific region. The mechanism requires backends to trust one another to function properly. GAEN [120] is capable of providing tracing internationally for all region-specific apps that build on its API.

3.5.9 *Performance Considerations*

The performance is a relevant aspect for estimating the feasibility of a DCT system and is closely linked to the adoption rates [189, 299]. Most DCT approaches require large server storage, either to do risk assessment or to hold data that will be transmitted to the users. As uploaded data becomes irrelevant after 14 days for the purpose of contact tracing due to the incubation period of Covid-19, data retention periods can be short.

Regarding client-side storage, the authors of DP-3T [305] come to the conclusion that their various approaches require between 4.8 MB to 6.9 MB of storage for received pseudonyms. This assumes that received data has to be stored for 14 days, that keys are 48 bytes long, and that an estimate of 140,000 different observations are made in that time. Users generally also have to store their pseudonyms of the last 14 days, but depending on the scheme this data lies in the range of a few kilobytes to megabytes.

Requirements on bandwidth greatly depend on the design of the DCT system and are influenced by how often data is downloaded, how large the downloads are, and the number of diagnosed users. For scalability purposes, a content delivery network can be used [305].

Cryptographic operations are computationally expensive for mobile devices. One solution to this issue while ensuring devices remain usable is outsourcing the generation of pseudonyms to the cloud [142]. Here, local computation is traded for bandwidth. Another option is to use symmetric cryptography instead of the asymmetric alternative to reduce the required CPU cycles for generating a pseudonym [249]. Expensive functions that require costly HMAC operations should also not run too often.

Running a smartphone application in the foreground can drain the battery. As we have seen earlier in Section 3.3, running BLE scans in the background required special permissions and changes to the operating system. Otherwise, the battery is drained or scanning does not work at all in the case of iOS devices. Google and Apple are the two main providers for smartphone operating systems worldwide [216]. Since they have implemented their own background API following a broadcast-based design, any other application not using this API effectively drains the battery [237].

3.6 PRIVACY AND SECURITY CONSIDERATIONS

Security and privacy considerations play an important role in the design of DCT systems. In the following, common threats and potential solutions are summarized. First, attacks and defense mechanisms concerning BLE-based proximity detection are discussed. Afterwards, several types of attacks are presented that focus on deanonymizing the source of a warning and metadata leakage is analyzed. Finally, countermeasures are evaluated both from a design perspective as well as from an operational security point of view.

3.6.1 *Attacks on the BLE Layer*

This section discusses problems with and attacks against BLE in more detail.

Jamming

Companies or individuals wanting to prevent contact tracing on their premises can block the exchange of pseudonyms by jamming the parts of the radio spectrum used by BLE. This attack is easily mounted with additional equipment, as shown by Xu et al. [318], and cannot be mitigated [17].

Storage and Power Drainage Attacks

Another simple attack targets the exhaustion of battery power and storage of the end device by sending large amounts of BLE messages [144]. This might make the DCT system unappealing to users, hindering widespread adoption [173, 306]. This attack is easy to mount as only devices capable of sending BLE messages are needed, as shown by Chen and Hu [72]. Uher et al. demonstrated that a denial-of-sleep attack can be used to drain the battery of BLE receivers [306]. An additional antenna to increase the attacker's reach can make this attack more effective. One solution that DCT systems can use is filtering incoming broadcasts and blocking misbehaving senders [142]. GAEN [141] takes a sample of beacons at least every 5 minutes. The service responsible for handling received advertisements is specifically designed to be able to deal with large volumes of data. This is also relevant for when the user visits public spaces. GAEN also proposes using software and hardware filters to remove duplicates and deal with the case of many advertisements. To circumvent the duplicate filter, an attacker would need to invest resources and change their MAC address for every packet.

Replay and Relay Attacks

An attacker who wants to make people believe that they are at risk can record pseudonyms and replay these BLE messages. BLE advertisements can, for example, be collected at high-risk areas like a testing center and then be broadcast at a different. This attack allows to target locations frequented by a certain person or demographic. It has been argued that replaying a single pseudonym might not be sufficient to surpass the threshold duration and be counted as close contact. Using a dedicated antenna, the attacker can receive advertisements within 20-100m range [305]. To limit the impact of replay attacks, most approaches [68, 161, 170, 249, 305] encode the epoch of the encounter in the transmitted pseudonym. DCT systems where a server assigns pseudonyms can check when an encounter was recorded and whether the recorded pseudonym was actually in use at that time [142, 161, 247]. Broadcast systems allow users to check themselves if they recorded a corresponding encounter for this time slot. Pseudonyms can also be cryptographically linked to their epoch [305]. Some approaches include the sender's pseudonym at the epoch of the encounter or a distinct shared key to allow the receiver to do a similar check [40, 1]. This requires loosely synchronized clocks, but even deviations of several minutes are acceptable.

The situation is different when the attacker relays the collected pseudonyms during the same epoch in which they were collected. It has been shown in a real-world scenario that relay attacks, sometimes called wormhole attacks, are feasible in GAEN [44]. A mitigation proposed by Vaudenay was to switch from passively exchanging pseudonyms through broadcasts to an active protocol [310]. As discussed in Section 3.3, an active exchange of messages is less secure than one-way communication where users send and listen for advertisements as it opens the door for new types of attacks against the end device [40]. Energy consumption also increases in a scenario with active BLE communication.

Some works [144, 248] propose using coarse (GPS) location data in the broadcast of the pseudonyms, allowing the receiver to figure out if the sender is close. This can be improved by introducing a message authentication code to prove the authenticity of the geo-location encoded in the BLE message [249]. The BLE message can also indicate that no location information is available by using a specific pseudonym. If the majority of users do send location information over BLE, relay attacks are mostly mitigated.

Linking Advertisements

When an end device advertises itself, a MAC address is also part of the transmission. This MAC address changes regularly. To ensure that linking different pseudonyms of the same person is not feasible, it is

vital that the MAC address changes simultaneously with the pseudonym. This feature requires support by the operating system [142, 305]. It has since been implemented by Google and Apple [140]. The DP-3T consortium did experiments to verify that this attack vector is mitigated [16].

However, by measuring the time between the announcements of pseudonyms, an attacker can determine which successive pseudonyms belong to the same person. A simple solution against this issue is synchronizing the switching of pseudonyms between all users [68]. Since this requires somewhat synchronized clocks in all end devices, it has instead been proposed to add jitter to the intervals between announcements [144].

Another point when trying to mitigate linking attacks is to consider the RSSI data. These proximity measurements allow an attacker to determine if two successive pseudonyms originated from the same approximate location. Gvili [144] proposes to have senders vary the signal strength in a way that makes it difficult to deduce the location of a user from only a few samples. No DCT system known to the author of this thesis takes measures against this variant of linking attacks.

Location Tracking

A passive eavesdropper might listen in on BLE advertisements and collect pseudonyms over BLE. These can then be used to track and deanonymize users. One way to do this is by continuously linking BLE advertisements and tracking users over a network of BLE scanners. This requires an attacker to have financial resources to install the required infrastructure in highly frequented public places. Infrastructure for BLE sniffing already exists in some areas due to digital billboards being equipped with BLE sensors [214]. Baumgärtner et al. [44] conducted experiments on how to track healthy and diagnosed users of the German Corona-Warn App, which builds on the DCT API of Google and Apple [141]. They were able to derive a coarse location history for diagnosed users. This attack works especially well for DCT systems where past pseudonyms of diagnosed individuals are published to allow local risk scoring. This means all broadcast DCT systems are exposed to such an attack. One mitigation against this type of attack is secret sharing [4] beacons [305]. Instead of advertising the pseudonyms, only fragmented shares of pseudonyms are broadcast. The other side must collect a certain number of shares to deduce the sender's actual pseudonym. This means that the pseudonym cannot be reconstructed if a user is just passing a BLE scanner. Therefore, it becomes difficult for publicly located BLE receivers to collect meaningful pseudonyms from people simply passing by. A disadvantage of secret-sharing pseudonyms is that some contacts of sufficient duration might not be recognized [52]. This can be resolved by making time slots of sequential pseudonyms overlap

[4] *Secret sharing is a method for splitting information into mutliple parts which hide the content. The information can only be reconstructed if all or a subset of shares are combined* [291].

such that two pseudonyms are always advertised at the same time. The device address must be considered when secret-sharing beacons to ensure tracking is not possible.

Another option for an adversary trying to track users that do not require continuous tracking is leveraging weaknesses in the BLE protocol or the transmitting devices. An attacker can, for example, fingerprint the properties and imperfections of a BLE chip and use this information for to reidentify users [133].

Active BLE Eavesdroppers

An attacker might not be satisfied with passively collecting pseudonyms and instead equip BLE devices in public spaces with the targeted contact tracing app. This way, a passing user will collect a pseudonym originating from the attacker's BLE device. Since public places are usually crowded and most DCT systems change pseudonyms regularly, detection is unlikely. Even worse, if security cameras are equipped with DCT applications, exchanged pseudonyms can be linked to surveillance footage. This makes diagnosed individuals easily deanoymizable at a later point in time using corresponding video recordings. This attack – with or without surveillance footage – is slightly more complex to mount than the one presented in Section 3.6.1. DCT approaches where users do not learn which pseudonyms from their history of encounters belong to a diagnosed person are safe against this attack. Generally, all approaches discussed in this chapter labeled as broadcast systems are vulnerable to this attack, as well as some message-based approaches. The secret sharing of beacons helps users who are at the location only for a short period of time. The number of shares is an important parameter to consider, as more shares mean higher privacy but might harm utility.

3.6.2 *Deanonymizing Source of Warning*

Apart from attacking the physical BLE layer, an attacker can also try to gain sensitive information by using the time of encounter to determine who is the source of a warning message and, therefore, diagnosed.

Leakage through Time of Encounter

Most client-side DCT systems allow a user who has received a warning to learn at what time they have encountered the diagnosed person. A user can use their memory to reidentify the diagnosed person they have been in contact with. Some designs give the exact time of the encounter and its duration, while others only provide the epoch. The shorter an epoch, the easier this attack becomes. While this endangers the privacy of diagnosed users, this information is also a useful tool for

users to do sanity checks on warnings [249]. In Section 5.4, we present a privacy-preserving DCT protocol that does not suffer from this leakage.

One Contact Attack

Assume an attacker wants to find out if a target person Tiffany will be diagnosed at a later point in time. The attacker could create a new account to register an encounter with her. If a risk notification is sent later for this account, the attacker knows that Tiffany triggered it. One way to mitigate this attack would be to make creating a new account difficult, for example, by installing CAPTCHAs [5] or tying participation in the system to a phone number.

A solution to combat this issue that is simple to implement for server-based DCT approaches is probabilistic notifications [161]. Here, for a small percentage of requests the server receives from clients for risk scoring, a warning is sent independent of the risk score. This increases the false positive rate but provides plausible deniability.

Another solution applicable to all types of DCT is to ensure that a user is always protected by k-anonymity [144]. If less than k distinct BLE advertisements are detectable, end devices can create cover traffic to make it look like more users are in the general area. An observer will not be able to determine which transmissions come from which users, especially if the signal strength is varied.

To increase operational security, it has been proposed to employ remote attestation [12, 52]. Smartphone operating systems allow backend servers to verify the integrity of devices and applications that want to communicate with them by using Google SafetyNet or iOS DeviceCheck. These mechanisms allow the identification of altered apps, which are needed for the execution of a one-contact attack if there are other users around.

Rate limiting the number of queries that can be sent by a user in DCT systems that rely on queries to the server [304].

[5] *Short for "Completely Automated Public Turing test to tell Computers and Humans Apart". Usually a type of visual challenge response test.*

3.6.3 *Metadata*

An important aspect of operational security is to check whether metadata can leak information that must remain secret.

IP ADDRESS LEAKAGE Many DCT systems rely on the IP address not to be leaked when communicating with central infrastructure. Users of a system where risk assessment is done on the server might have an interest in not revealing their identity directly to the server. In broadcast-based systems, users might not want to reveal the fact that they participate. Also, depending on the authentication mechanisms, users might want to ensure that uploaded data (like past pseudonyms) is not linkable to their identity. For this purpose, anonymization networks like

Tor [100] or mix networks [88, 98] can be used. If users use such a network when communicating with the server, it will not learn their real IP addresses (and thereby their identity) as they are hidden by a cascade of proxies. While Tor-like anonymization infrastructure is vulnerable against timing attacks conducted by adversaries capable of monitoring large parts of the network [240], mix networks do not have this drawback but are slower at delivering messages. In some cases, IP Address leakage can be mitigated by using an independent messaging service that transmits encrypted messages [46].

Another solution to this issue relies on the assumption that the healthcare provider (for example, the facility where the user got tested) knows the diagnosed user's identity and can be trusted [304]. A diagnosed user can freely communicate with the healthcare provider and upload their encrypted data there. The healthcare provider can collect data from multiple diagnosed users and shuffle it before uploading everything to the server responsible for DCT. It works as an anonymization proxy and is, therefore, not allowed to collude with the DCT server.

TRAFFIC ANALYSIS Anonymization networks do not only hide IP addresses but also stop an attacker who observes the network from finding out who is diagnosed or at risk. On the downside, they are known to have performance and scaling issues [18]. It has been argued that the current Tor network is not equipped to support the expected user basis of a DCT system [247]. Therefore, mechanisms are proposed that leak the user's IP address but defend against network observers. To ensure a network observer does not learn if an upload contains real data which indicates that the sender is diagnosed, one solution is to regularly upload dummy data [12]. If a user is diagnosed and uploads their real data, the app must continue downloading keys and making fake requests. In case data is uploaded during the daily warning check, uploads can be hidden in this upload. Not securing this path allows a network observer to identify diagnosed individuals.

LEAKAGE THROUGH UPLOAD TIMING Timing is another type of metadata that allows to derive information about users. When uploading data that should not be linked by the server like warning messages originating from different pseudonyms, it is necessary to also induce jitter. This is discussed by Robert [161] to break the link between two uploads from the same diagnosed user. The authors also consider mix networks and additional servers with secure hardware modules for this purpose. Additionally to jitter, mix networks, and onion-routing, cover traffic is helpful for client-oriented DCT systems to defend against this type of linking attacks [40]. It becomes unclear for the attacker which data is real. This method can only be applied to systems where the server cannot tell real data from decoys. Similar to other attacks

based on metadata, an independent anonymization proxy can solve this problem as well [304].

3.6.4 *Hacking, Backdoors, and Malware*

DCT systems generally rely on apps being installed on the user's smartphone. Like in any kind of IT environment, both underlying hardware and software can be vulnerable. Therefore, regular updates are mandatory to ensure security and privacy. To guarantee that no other installed applications can spy on the DCT app, it has been suggested that employing Trusted Platform Modules (TPM) [6] would help [310]. Remote attestation mechanisms available in most smartphones are also useful to detect hacked devices [12].

[6] *A TPM is a type of computer hardware that is physically isolated from other parts and can provide secure storage of keys.*

However, hackers can also attack servers directly and use log files to identify diagnosed users by the IP address of their upload. To prevent this kind of privacy leak, it has been recommended not to maintain logs that might leak the identity of diagnosed users [266]. Relying on anonymization networks and dummy traffic also hides information in log files which might be of interest to hackers [12, 266].

Users' trust is an essential building block of DCT systems. It has often been argued that making code open source is a requirement to ensure that a DCT system is trustworthy [52, 75, 153]. Having code freely accessible allows independent security researchers to check that no backdoor has been implemented and that the app does not contain malware. Open source code is available for various DCT apps discussed in this chapter, for example BlueTrace [142], PePP-PT NTK [247], Robert [161], DP-3T [305], Hashomer [249] and Covid-Watch [38]. Additionally to having code freely available, independent audits are necessary to ensure that the published code is used to run the backend servers or to build the application. Here, the usage of a trusted execution environment on the server side can help to prove to users that the source code running on the server is the same as the one that is openly available [247].

3.7 DISCUSSION

In this section, we discuss the societal factors of DCT and take a look at how effective BLE-based DCT was at combatting the pandemic.

3.7.1 *Adoption and Public Perception*

For DCT to be successful, widespread adoption is necessary. Simulations evaluating the effectiveness of DCT use adoption rates from 40% [243] and 53% [192] to 56% [155] of the population. Some authors have suggested that these numbers should not be understood as hard limits, as apps do not become useless at lower adoption rates but rather

less effective [157]. In another modeling study, Kretzschmar et. al. [191] find that DCT is more effective than manual contact tracing, even if only 20% of the population uses the tracing app. This follows from the shorter delay of notifying contacts compared to the manual approach of interviewing a patient and then calling their previous contacts via telephone.

To improve effectiveness, some states have made using the local DCT app a requirement for using public transit and participating in public life [90]. For most European countries, such measures sparked serious concerns regarding civil liberties and discrimination against people without a suitable smartphone [19, 115, 195]. If installing DCT apps is voluntary, public perception is an important factor impacting adoption. The willingness to adopt contact tracing apps was strong at the beginning of the pandemic, as shown in the cases of the USA and Germany [302]. Educating users about the benefits of DCT apps for themselves and society positively impacts acceptance [302]. Intrinsic motivation for using a DCT app is an important factor for its penetration. In some regions, the adoption of DCT and using the local app was linked directly to the hope of users that regional rules like lockdowns and contact would be lifted faster [50, 124, 189].

Security and privacy concerns regarding the DCT app hinder adoption [28, 189, 283, 299]. For example, apps that allow uniquely identifying people are considered less trustworthy [189]. It has been shown that systems perceived as surveillance measures are seen as less trustworthy and people are less inclined to install the corresponding DCT app on their devices [58]. Toch and Ayalon [299] showed in the case of Israel, that when mass surveillance measures are in place for contact tracing, people are less inclined to trust and install any voluntary contact tracing apps. In general, the attitude toward the government significantly influences the willingness to use contact tracing apps [28, 189]. Another factor that decreases the willingness is a fear of data misuse by third parties such as companies or law enforcement [189, 283].

However, the "paradox of privacy" also applies to DCT [299]. While people do have privacy preferences they might not act in accordance with these preferences. Toch and Ayalon [299] explain this discrepancy with the difficulty of assessing privacy risks of different DCT architectures.

3.7.2 *Usability*

Usability aspects have to be taken into account when talking about the adoption rates of DCT systems. One usability requirement voiced by both app developers and users is that the DCT app should not drain too much power [142, 189, 299]. Users also should not be disturbed by the application. It should, therefore, be capable of running in the background without needing to be opened regularly [142]. A similar

requirement is the automated processing of data without user interaction, as well as refraining from having users perform manual tasks that are error-prone and time-consuming, such as entering a long number read out over telephone [18].

A large factor regarding the usability of a DCT application is its graphical user interface. Many DCT applications only gave feedback to users when infection warnings were shown. According to Kowalewski et al. [189], this app behavior might be misinterpreted by users, causing them to believe that the app is not working correctly. The step of uploading data after a positive test was also a cause for problems. While this step is essential for delivering warnings to potentially infected users, it was often abandoned in the process [59, 89, 110]. A usability study of the Dutch DCT app found that users were confused about the upload procedures [48].

Analyses of the adoption of contact apps showed that the apps were more widespread in some parts of the population than others [48, 283, 299]. To reach everyone, app interfaces must be accessible and inclusive for older generations, people with language barriers, and those lacking technological literacy [48, 283]. For the same reasons, compatibility with older smartphone models is essential as it ensures that disadvantaged groups, such as people with lower economic status, can participate in smartphone-based contact tracing [283].

Throughout the pandemic, people had to install multiple apps for different purposes, such as contact tracing, location check-ins, and vaccination passports. Kowalski et al. [189] found that people would prefer to install and use one app instead of multiple. This might again raise security and privacy concerns [283].

3.7.3 *Effectiveness and Efficiency*

Besides technical issues, questions of utility played an essential role in the decision to use (or not use) a DCT app [189]. Especially false alarms seemed to cause frustration. For example, in the summer of 2021, the number of notifications sent by DCT apps was very high [182]. This so-called "pingdemic" [182] can be partially attributed to many cases and an increased number of social contacts due to a partial reopening of public places.

People infected with Covid-19 are already infectious before showing symptoms [228]. As a result, a large fraction of transmissions occur before the person in question becomes aware of their infection. Modeling studies concluded that manual contact tracing would have not been able to contain the virus due to transmissions by infected people without symptoms [124]. DCT promises that by automating this process, new infections are detected faster [124, 203]. Additionally, it allows notifying contacts that would be missed in manual tracing, for example, in public places. In some regions, manual contact tracing broke down

entirely during the pandemic due to high infection rates and a shortage of tracers [97, 200]. In these cases, DCT filled the gap left by manual tracing.

However, the effectiveness and impact of DCT on the course of the pandemic are disputed. One issue is the efficiency of the testing and tracing pipeline. Let us assume a person gets tested on the onset of symptoms. After the test comes back positive, the now-diagnosed person can request tokens to upload data (for example, via a telephone hotline) and notify potential contacts. If this process is too slow, the notified contacts might have already infected other people during the latent period. Alternatively, the notification becomes useless as the contacts already show symptoms. Kretzschmar et al. [191] found that contact tracing would not be effective at stopping the spread of Covid-19 if the delay between testing and notification is three days or longer. Cencetti et al. [66] determined that a delay of two days in combination with other measures would be effective. Nevertheless, in their model, a tracing delay of three days, even in combination with strict measures, was not enough. The importance of the testing-tracing pipeline can be seen in the case of Switzerland, where the tracing effectiveness temporarily stagnated as the local health authority could not hand out upload tokens due to a shortage of staff [89]. This issue was solved by involving third-party services.

Analyses on the British NHS app [182, 317], the SwissCovid app [42], the Norwegian Smittestopp app [111], the German Corona-Warn app [110], a controlled experiment study in Spain [269], and a simulation based on pessimistic parametrization [59] confirm that DCT based on proximity detection with GAEN was successful in limiting the number of infections and effects of the pandemic. It has been suggested that DCT is not a replacement for other measures, such as manual tracing and mask mandates, but an extension [59, 66, 317]. DCT based on BLE only allows direct contact to be traced. This seems to be sufficient as a modeling study by Cenetti et al. [66] found that tracing second-degree contracts (which is impossible for client-side DCT applications) does not improve efficiency.

The course of the pandemic was influenced by the infectiousness of the most prevalent virus variant, mask mandates, physical distancing, lockdowns, how efficiently people were isolated once diagnosed, immunity rates, super-spreader events, test availability, and many other factors. For these reasons, the utility of a DCT application in use needs to be reevaluated regularly. An app that was useful in a situation with a high mortality rate and no vaccine available might not be suitable in an epidemic situation with a partly vaccinated population [59].

3.7.4 *Other Digital Tools for Pandemic Control*

DCT for proximity detection is not the only type of tool for finding new infections or mitigating the spread of a virus.

For example, the Chinese Health Code system used big data to determine the infection and transmission risk of users [201]. The system utilized personal data such as GPS traces and hospital records [201], but also data regarding the general population [78]. The resulting risk scores were not directly linked to the exposure to infected people but rather served as a tool for preventive population control.

During the pandemic, many Covid-19 infections were linked to so-called super-spreader events [69, 313]. In these cases, a single individual, often without symptoms, transmits the virus to a large number of others present. These occurrences are more likely in indoor settings or in locations with tightly packed crowds, such as rehearsals [150], weddings [217], political events [236] or restaurants [126]. As a result, systems for DCT through *presence tracing* were proposed. Here, not the proximity to an infected individual is relevant but whether a person stayed in the same public or quasi-public space. Some jurisdictions did require operators of such locations to collect contact information from visitors [270]. However, manual tracing using this data is time-consuming and error-prone as written information might be unreadable, incorrect, or incomplete. The process was quickly digitized to allow users to *check-in* into a location by scanning a QR code [94, 213, 222, 286]. Privacy guarantees of these presence tracing apps and proposals varied. Chapter 6 presents an approach for privacy-preserving presence tracing that extends the proximity detection of GEAN.

Other types of tools for pandemic control include apps for symptom tracker [78, 149], quarantine enforcement [149, 189], and vaccination passports [95]. In some cases, multiple functions were combined in one app. The German Corona-Warn-App provided proximity-based DCT and a vaccination passport [95].

3.7.5 *Misuse of DCT Data and Systems*

One hope connected to the deployment of DCT systems has often been that lockdowns will be shortened and life will return to normal [50, 124, 189]. However, what means are acceptable to achieve this goal?

DCT apps were initially met with distrust that app data might be misused. One concern was that app developers might use data collected for contact tracing for their economic gain [283]. This fear does not seem to be entirely unfounded. For example, after its use as check-in app for locations such as restaurants and cafés, the German LUCA app tried to find a second life as a payment service for the gastronomic sector [94]. Another example of the commercialization of Covid-19 data is the Chinese Health Code system. Here, mini-apps were integrated into

the platforms of companies such as Tencent and Alibaba to determine the risk of individuals based on big data. The storage of sensitive data on commercial platforms and the potential use has raised criticism [78, 201]. This public-private partnership also strengthened the position of the respective companies. The issue of private companies encroaching into the area of state responsibilities and introducing quasi-standards has also been criticized with respect to the GAEN contact tracing design as it made all other BLE proximity-based designs infeasible [201, 312]. DCT data can also be misused due to illegal access or hacks. Following a hack of the Chinese Health code system, the private data of celebrities was sold online [319].

Even more worrying than the economic exploitation is the fact that the police and intelligence services also accessed the data collected for contact tracing purposes. In Singapore, data collected by the TraceTogether app, a centralized DCT system, was used in criminal investigations despite previous promises during the app rollout that this would not be the case [71, 218]. While the app has been removed in the meantime, not all data was deleted [71]. Similarly, data collected by the German LUCA app was surrendered to the police in a criminal investigation [218, 260]. Australian police also used data collected for contact tracing on several occasions. In various countries, the resulting public outcry caused lawmakers to add additional, specific privacy protection for this type of data [71, 218]. While in all these cases, the access to contact tracing data was legal, they hindered the overall goal of contact tracing by hampering the app adoption [189, 283].

Some states used the pandemic to employ systems that might later prove to be capable of dual-use [147, 149, 186, 289]. In Israel, during the first and second waves, existing but up to that point untested surveillance measures were used to discover new infections [147, 218]. In some countries, such as Singapore and China, using the local contact tracing app was mandated or quasi-mandatory. The usage of Covid-19 data for crowd control has been reported in Israel, India, and China [61, 78]. In China, data in the Health Code system has been altered illegally to stop protests [259].

It has been argued that surveillance for contact tracing normalized the use of these technologies also in other areas [61, 201]. Li et al. [201] note that the Covid-19 pandemic accelerated the digitalization of governments. Some existing apps were expanded step by step, changing the core functionality. This development is often called function creep, which can be observed in the case of the Chinese Health Codes system. In some regions, the system was extended for purposes such as enabling access to public services and health care [78].

3.8 CHAPTER SUMMARY

This chapter introduced the concept of digitally tracing contacts of diagnosed individuals in case of an epidemic or pandemic by using proximity data. The goal of DCT is to relieve pressure on the manual tracers, reduce the spread of the virus, and, thus, slow down the pandemic by speeding up the detection process and informing people at risk of infection as early as possible when they need to quarantine. Many new ideas for DCT approaches were presented in the first years of the Covid-19 pandemic. This chapter explained why BLE emerged as the most used sensor type for proximity-based DCT systems and discussed a systematization of BLE-based systems presented by research, industry, and governments. Based on this systematization, typical attack vectors were described. We identified common challenges to DCT and presented a wide range of solutions with regard to functionality issues as well as threats to privacy and security. With a retrospective view of the Covid-19 pandemic, we analyzed the literature on the societal risks of contact tracing and its effectiveness during the fight against Covid-19.

As shown in this chapter, privacy concerns are an important factor in the adoption of DCT apps. Approaches that store data in plain text on a server are prone to misuse by hackers or by law enforcement. It is, therefore, essential to protect users' privacy through technical guarantees. The following chapter, Chapter 4, will take a look at how such guarantees can be achieved through client-side risk detection with direct messaging. Chapter 5 examines the use of cryptographic protocol as a means to this end. As mentioned above, presence tracing is a challenge adjacent to proximity-based DCT. When data regarding visited locations is used for tracing, different methods have to be used to protect the sensitive information of both diagnosed users and everyone else. The last chapter of this part of the thesis, Chapter 6, discusses how super-spreader detection through presence tracing on top of GEAN can be accomplished in a privacy-preserving manner.

4

CLIENT-SIDE RISK ASSESSMENT THROUGH DIRECT MESSAGING

As we have seen in the previous chapter, client-side risk scoring for Digital Contact Tracing (DCT) ensures that a central authority does not learn sensitive information, such as who is at risk or who interacts with one another. *Direct messaging* is one of the ideas presented in the systematization in Section 3.4.5 that enables privacy-oriented solutions by handing over responsibility to the client. The concept was first presented by Cho et al. [75] in a preprint at the beginning of 2020. For direct messaging, users regularly create a new asymmetric key pair and use the public key as an ephemeral Bluetooth Low Energy (BLE) pseudonym. When a person is diagnosed, they place messages encrypted to the pseudonyms of their contacts into the corresponding postbox. Users need to regularly check postboxes belonging to their past advertised pseudonyms to see if a new message has arrived. One issue not discussed by Cho et al. is the aspect of authenticity. Users can try to cause panic by sending "I am infected messages" to many people without actually being at risk. Additionally, special attention must be paid to potential privacy leaks caused by metadata. If messages are only sent in the case of a diagnosis, an adversary can determine that all users who receive a message are at risk. For this reason, cover traffic is essential to ensure privacy.

In the following, two contributions are presented that provide solutions for these problems. The first proposal CAUDHT in Section 4.2 utilizes *Distributed Hash Tables* (DHTs) for a scalable postbox infrastructure. Additionally, *blind signatures* are leveraged to provide authenticity guarantees for messages. However, DHTs allow any party who is interested to listen to the traffic and analyze the metadata. For this reason, the second proposal Ovid in Section 4.3 simplifies and improves the design of CAUDHT. A main contribution of Ovid are the considerations regarding cover traffic. Related work for CAUDHT and Ovid are presented in the following in Section 4.1.

4.1 RELATED WORK

The two designs discussed in this chapter utilize the concept of client-side risk scoring as introduced in Section 3.4.2 in the previous chapter. The systematization focused on providing a broad overview. A more detailed explanation of the relevant systems is required to compare CAUDHT and Ovid against other works in the field. This section ex-

plains the designs of all client-side risk-scoring approaches mentioned in Section 3.4.2 [7].

[7] For details on server-based approaches to DCT that preserve privacy, the reader is referred to Section 5.2.

4.1.1 *Broadcast-based*

During the pandemic, broadcast-based systems had the most practical relevance due to the Google Apple Exposure Notification (GAEN) API provided in Android and Apple smartphones (see Section 3.4.3). However, broadcast-designs with different properties were also presented [8]. Similarly to direct messaging approaches like CAUDHT and Ovid, all broadcast-based designs require cover traffic to hide the identity of diagnosed users from network adversaries. However, only uploads from newly diagnosed users need to be masked.

[8] See Table A.2 in Appendix A for a tabular overview of the mentioned broadcast and targeted broadcast-based designs.

DP-3T

A proposal that presents several closely related designs for broadcast models is DP-3T [305]. The so-called *DP-3T low-cost design* is similar to GAEN and allows users to use an individual seed to derive a daily key from which pseudonyms are derived. A major problem with this approach is that the pseudonyms of diagnosed users become linkable, allowing for potential tracking and easier deanonymization of diagnosed users. To mitigate such attacks by curious users or eavesdroppers, DP-3T developed a second approach called the *unlinkable design*. Here, for each epoch, a cryptographically independent pseudonym is generated. When a person becomes infected, all pseudonyms are uploaded to a server that stores them in a global hash table [305]. Users will download the hash table regularly and check if any of their past encounters cause a hash collision. To ensure that the failure probability of the hashing process remains low, the server creates a new, empty table when necessary [73]. When data is uploaded, the server – and thus the health authority or the service operator – learns the past pseudonyms of a diagnosed user but not with whom they interacted. This is not the case for CAUDHT as the server is distributed over a DHT. In the case of Ovid, this leak can be mitigated by planning uploads and breaking the link between uploads.

Hashomer

Pinkas and Ronen proposed a similar broadcast system called Hashomer, which relies on an elaborate key derivation mechanism [249]. Similarly to GAEN, the keys advertised at different epochs of the same day are derived from a daily key and unlinkable. The server can either broadcast the pseudonyms of diagnosed users or the daily tracing key and increase performance in favor of privacy. Hashomer's key derivation mechanism for pseudonyms allows users to prove their exposure

in case of an encounter. To ensure that an adversary does use multiple devices that use the same pseudonyms, Hashomer generates a commitment key at installation time, which influences all other key derivation mechanisms.

In Hashomer, the pseudonyms for one day are cryptographically linkable by the server. The proposed systems CAUDHT and Ovid break this link between pseudonyms.

4.1.2 *Targeted Broadcast*

Systems that use targeted broadcast work similarly to broadcast-based ones with the main difference being that the broadcasted data has one dedicated receiver. Targeted broadcast systems suffer from similar weaknesses as simple broadcast-based proposals. Uploads from diagnosed users need to be hidden from a network observer through cover traffic or other anonymization methods.

ConTra Corona

An example of the targeted broadcast approach is ConTra Corona [52]. For each epoch, users derive a secret and a public pseudonym from a newly generated seed. The public pseudonym is advertised over BLE, while the seed is uploaded to a so-called matching server. If a user tests positive for Covid-19, they encrypt all relevant recorded pseudonyms with the public key of the matching server and have the health authority forward the data. The matching server decrypts the data, looks up the corresponding seed, marks it as infected, and generates the corresponding secret pseudonym. Users can either query for their secret pseudonyms or the matching server publishes them regularly. In ConTra Corona, users at risk will only learn during which time period an encounter occurred. The security of the system relies on the assumption that the health authority and the matching server do not collude. In contrast, CAUDHT and Ovid do not require such an assumption.

Pronto-C2

Another example for targeted broadcast is Pronto-C2 [40]. Here, users derive a shared key from the ephemeral pseudonyms continuously advertised by users. For this purpose, a Diffie-Hellman key exchange [99] [9]is used, so only the two parties can identify the shared key. Since pseudonyms are rather long, they are uploaded to a bulletin board and only a link to the pseudonym is transmitted over BLE. If someone is infected, the shared key is published and distributed to all users. The authors propose to use a blockchain to ensure that no data can be deleted.

[9] *This key exchange allows two parties to jointly derive a secret key by exchanging messages over a public medium.*

From a cryptographic standpoint, the targeted broadcast-based design Pronto-C2 [40] ensures only the addressee is capable to determine that a warning messages is targeted at them. However, the authors of Pronto-C2 did not consider metadata leakage in their design. Due to their length, pseudonyms are stored on a public bulletin board. If no additional measures for hiding a user's IP address are taken, the storage server will learn who interacted with one another by monitoring who reads which pseudonyms from the bulletin board. This means the health authority, the service operator, and network operators might learn the social graph. But, they will not learn who is at risk. Users of the system will know which shared key belongs to which encounter and thereby be able to deanonymize infected users using their background knowledge. By utilizing the idea of postbox channels as introduced in Ovid, reads to the bulletin board could be hidden in cover traffic.

Covid-Watch

Covid-Watch [38] is another form of targeted broadcast. An early proposal of of the Covid-Watch project required users who test positive to not only upload their own pseudonyms but also those they have recorded. These data tuples are then broadcast to all other users who check locally if they have a corresponding encounter stored.

Whisper

Unlike most DCT designs, the Whisper Tracing Protocol [212] does not only rely on BLE but also utilizes active Bluetooth connections. Mobile devices scan for other compatible BLE devices and initiate an active connection to derive a session key. In case of a diagnosis, these keys are published. The matching system can be run both in a central and decentralized manner, i. e., the matching can be done on the server or the end devices. This allows a trade-off between user privacy and the server being able to learn about the epidemiological parameters of the disease. Similarly to CAUDHT, the authors propose to distribute the server architecture over the Interplanetary File System (IPFS), a peer-to-peer network for file sharing. By requiring an active connection to exchange pseudonyms, the Whisper Tracing Protocol is susceptible to attacks against the end device.

4.1.3 *Direct Messaging*

Both CAUDHT and Ovid build on the idea of *direct messaging* for DCT. Similar approaches relying on this concept will be presented in the following [10].

[10] *See Table A.3 in Appendix A for a tabular overview of all direct messaging-based designs.*

TraceSecure

The privacy of TraceSecure [46] relies on multiple non-colluding parties. These are the health authority, the government, and in some cases a messaging service. When joining the system, users have to (anonymously) send their seed used for deriving ephemeral BLE pseudonyms to the government. In return, the user is given a static ID, which they can use to check with the messaging service if new messages have arrived. When a user is diagnosed, they notify all past contacts individually by having the health authority relay encrypted messages to the government. Each message contains an observed pseudonym. Since the government knows the seeds from which pseudonyms are generated, it can derive which static IDs need to be warned. Via the messaging service, it distributes encrypted warnings to the designated receivers. This system requires cover traffic on the path from the government to the user, so the messaging service and a network eavesdropper do not learn who is diagnosed. Since the health authority holds the seeds for all users, it can derive a user's current advertised pseudonym and use this information for tracking. However, it does not learn who received a warning and is at risk. The government learns the static ID of who is at risk but not who they have been in contact with. The privacy of users relies on the server not being able to link these static IDs to real identities. Users in this system only learn that they are at risk but no additional information and can, therefore, not conduct meaningful attacks.

Cho et al.

As mentioned above, the postbox system of Cho et al. utilizes pseudonyms transmitted via BLE to address and send messages to contacts [75]. These pseudonyms are the public part of an asymmetric key pair. Warning messages to a contact are encrypted with this pseudonym and will not indicate who has sent it. This means the user at risk cannot deanonymize infected users. To ensure that the server, and thereby the health authority or a service operator, cannot link real identities with postboxes, Cho et al. require requests to the server to be sent through a network of proxies. Additionally, users not only send messages to others when they get infected, but they also send messages stating that they are still healthy. The server only sees one user placing messages in a postbox but cannot decrypt this message and find out if the message is a warning or a decoy. As mentioned above, an issue of this proposal is that users at risk can not verify if the message they have received originated from someone who is actually infected. Cho et al. also do not discuss how often decoy messages need to be sent or how to mitigate spam and Denial-of-Service (DoS) attacks.

4.2 CAUDHT: DECENTRALIZED CONTACT TRACING USING A DISTRIBUTED HASH TABLE AND BLIND SIGNATURES

This section is based on previous work with Samuel Brack and Björn Scheuermann presented at the IEEE Conference on Local Computer Networks (LCN) 2020 [1]. The considerations on ECC key and DHT scalability for mobile devices as well as the discussion were added for this thesis.

A common approach for ensuring the authenticity of messages are cryptographic signatures. However, in the case of DCT with direct messaging, this step can leak the identity of the recipient. By checking postboxes, the singer can identify messages it has signed early and, thereby, learn who is at risk. It can also learn who interacted with one another. Here, blind signatures can be used to ensure a valid signature is provided while hiding from the message contents [70]. This section proposes CAUDHT, an approach that leverages this signature schema to provide privacy and authenticity for DCT with direct messaging.

The second contribution of CAUDHT is the proposal to use DHTs as postbox infrastructure. In settings where no centralized authority is desired, peer-to-peer networks are an excellent solution for sharing the responsibility of maintaining a service over many entities. DHTs are structured peer-to-peer networks that provide the functionality of a distributed key-value store. Each peer in the network manages a designated key space and keys are derived through hashing. DHTs are a scalable solution to deal with network bottlenecks, ensure availability, and mitigate metadata leakage that emerges from direct communication with a central server. Summarized, the main contributions of CAUDHT are:

- A design for DCT based on direct messaging.
- Introducing blinds signatures for message authentication in direct messaging DCT.
- Using a DHT as decentralized message exchange.

This section is organized as follows. First, the system design of CAUDHT is presented in Section 4.2.1. Then, details regarding pseudonym derivation (see Section 4.2.2), message construction (see Section 4.2.3), and the distributed messaging infrastructure (see Section 4.2.4) are discussed.

4.2.1 *System Design*

Like other DCT systems, the encounters in CAUDHT are registered by passively exchanging ephemeral pseudonyms over BLE with other users close by (see Section 3.3). In CAUDHT, each pseudonym is the public key of an asymmetric key pair. Elliptic Curve Cryptography (ECC) is used to derive these key pairs.

Let us assume Bob has had a high-risk encounter with Alice while she was not yet diagnosed. Their pseudonyms at the time of this encounter were P_{Alice} for Alice and P_{Bob} for Bob. When Alice is diagnosed, she

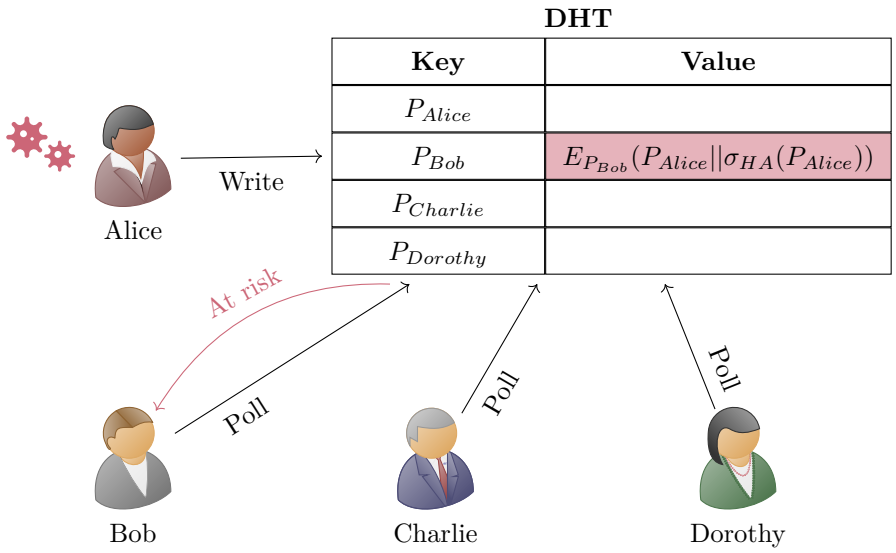


Figure 4.1: Writing and retrieving warning messages in CAUDHT. Alice sends a warning to Bob using his ephemeral pseudonym P_{Bob} as address. The message is encrypted to P_{Bob} and signed by the health authority (HA). Figure as in [1].

identifies all contacts with a high risk using her list of recorded pseudonyms and constructs an encrypted warning message for each of them. The warning contains her pseudonym at the time of the encounter. To prove her infection to these contacts, she retrieves a blind signature σ_{HA} from the health authority for each message. The format of her message to Bob is:

$$m = E_{P_{Bob}}(P_{Alice}||\sigma_{HA}(P_{Alice}))$$

Here, $E_k(c)$ represents the encryption of contents c with key k and $||$ is the concatenation function. She then places all messages in the DHT using the corresponding pseudonyms to address the postboxes of her contacts.

Like all other users, Bob regularly checks the postboxes corresponding to pseudonyms he used in the past by querying the DHT. After Alice uploads her messages, Bob will find that the postbox for the pseudonym P_{Bob} contains a message. To ensure the message is legitimate, Bob first verifies the signature of the health authority with its public key. If successful, he decrypts the message using his pseudonym and will learn that the user with P_{Alice} is infected and that he is at risk. See Figure 4.1 for a visual representation.

4.2.2 *Pseudonym Derivation*

The basic version of BLE allows advertising 31 bytes of information [152, 282]. However, the portion that can be effectively used to share key

information with others is smaller. The BLE extension for transmitting sensor data can be used for even longer pseudonyms. The extension splits transmitted data over multiple BLE packets, which has the disadvantage that subsequent packets can be lost as the risk of concurrent transmissions increases.

It is, therefore, preferable to fit all data to be advertised into on BLE packet. To this end, several optimizations can be applied. CAUDHT uses ECC public keys as pseudonyms. A ECC private key sk is a random integer, while the corresponding public key pk is a point on the elliptic curve created by $pk = sk \cdot G$ where G is a generator point on the elliptic curve. It is possible to compress an ECC public key to half its size by dropping the y coordinate and adding a sign bit. Apple's Offline Finding system leverages the `NIST P-224` curve and advertises a 28 byte ECC key with a single BLE package [152]. To do the bytes of the random BLE address are misused to transmit data. Offline Finding does not transmit the ECC point's sign bit and uses one byte to transmit the device status, which would not be required for our purposes. By using the random BLE address in a similar manner for CAUDHT, a 28 byte ECC public key and 1 bit sign information can be transmitted in a single BLE advertisement.

4.2.3 *Message Authenticity*

As mentioned above, signing messages protects users from malicious attackers trying to convince them that they are at risk. Using blind signatures ensures that the signing authority does not learn the message's contents and can not use it to iterate over postboxes and identify sender-receiver pairs.

Blind signatures for RSA [11] keys work as follows. The message author Alice has a message m and the public RSA key $pk_{HA} = (e, N)$ of the signer (here, the health authority). Alice blinds the message by computing $b(m) = c^e \cdot m \ (mod\ N)$ for some random but well-chosen value c. The signer uses its secret key $sk_{HA} = (d, N)$ and returns $\sigma(b(m)) = b(m)^d \ (mod\ N)$. Now, using the knowledge that $e \cdot d = 1 \ (mod\ N)$ Alice can compute:

[11] *A scheme for public-key cryptography named after its inventors Rivest, Shamir, and Adleman.*

$$\begin{aligned}\sigma(b(m)) \cdot 1/c &= (c^e \cdot m)^d / c \\ &= c^{e \cdot d} \cdot m^d / c \\ &= m^d = \sigma_{HA}(m)\end{aligned}$$

This provides a signature with key sk_{HA} over message m without the singer ever learning m. Blind signatures can either be linkable, allowing the signer to identify blinded messages from the same message author, or unlinkable. In the case of CAUDHT, unlinkable blind signatures are preferred to stop the health authority from learning how many signatures each diagnosed user retrieves.

Only diagnosed users should be able to retrieve blind signatures. For this reason, the health authority only signs messages if a token is presented that corresponds to a positive Covid-19 test. Various token mechanisms are discussed in Section 3.5.3.

A network eavesdropper observing the communication patterns of the health authority's signing server can learn who is at risk. This can be mitigated if diagnosed users hide their IP address using anonymization services such as Tor [301] or the Nym network [98]. Alternatively, cover traffic or domain fronting can be used to hide who retrieves a signature.

4.2.4 *Distributed Messaging Infrastructure*

DHTs are structured peer-to-peer networks that do not rely on a central entity for coordination. Different DHT protocols like Kademlia [221] or Chord [293] rely on different approaches to construct the overlay network. This results in different complexities when placing and searching content in the network. An advantage of DHTs is that there is no single point of failure. When nodes leave the network, the keys this node has managed are transferred to other nodes. Even in case of network churn, the contents of the DHT, such as messages, are still available for download.

For CAUDHT, it is assumed that all users of the system also participate as nodes in the DHT. This means that the DCT application needs to run on smartphones. Data transfers for mobile devices are often subject to charges and not unlimited. Additionally, reachability issues are a major concern due to network address translation, which is typical for mobile connections. The Internet connection of mobile devices is less stable than is standard for DHT nodes as their mobility is high. Evaluations on mobile peer-to-peer networks suggest that the proposed infrastructure of CAUDHT is indeed feasible [268]. Using a hierarchical DHT [12] can help to mitigate the impact of churn on the network [268]. Another concern when running a DHT application on mobile devices is that network traffic, such as keep-alive messages, drains the battery. However, various mechanisms are proposed in research to reduce the energy consumption of peer-to-peer systems such as DHTs [56].

[12] *Here, the DHT is established between so-called superpeers, which are characterized by a high bandwidth and long uptime. All others are leaf nodes connected only to superpeers.*

4.2.5 *Privacy and Security Considerations*

With regards to privacy and security, a problem with DHTs is that an attacker can perform a Sybil attack by participating in the network with multiple nodes. By placing Sybil nodes in specific key spaces of the DHT, the adversary can take over parts of the network. This way they can attempt to control all postboxes belonging to a specific user if they know the address of these postboxes, allowing them to monitor for new messages or actively remove messages. To mitigate Sybil attacks, running many nodes in parallel needs to bind resources on the attacker's

side. Solutions to this end are, for example, CAPTCHAs or requiring a phone number per user. Remote attestation, as used, for example, by the MobileCoin network [229], ensures that all nodes in the network run the same code. This can be used to ensure honest-but-curious behavior from the participants in the peer-to-peer network.

A threat to privacy resulting from relying on a DHT for distribution warnings is that no access control can be applied. Users regularly query the network to receive warnings for one of their past pseudonyms. However, any other user can also query for these pseudonyms but will not be able to read the message. This creates a side-channel where, if a message is returned, it is clear that the designated receiver might be at risk of being infected. As the entropy of addresses is large, an attacker cannot simply iterate over all possible postbox addresses. However, by listening to queries in the DHT or recording BLE advertisements at a location, the attacker can learn postbox addresses.

4.3 OVID: MESSAGE-BASED DIGITAL CONTACT TRACING

This section is based on previous work with Samuel Brack and Björn Scheuermann presented at the NDSS Workshop on Secure IT Technologies against Covid-19 (CoronaDef) in 2021 [4].

In the previous section, we identified that a major drawback of using a DHT as a distributed messaging infrastructure for direct messaging DCT. While providing scalability and availability, the privacy of users in CAUDHT suffers from the fact that anyone participating in the network can learn if users they have met have received a warning and are at risk. Introducing cover traffic is an option to mitigate such leakage. To assess the overhead of cover traffic in the case of DHTs, the trade-off between local storage per peer and the worst-case complexity for finding a key has to be considered. Common DHTs such as Chord [293] or Kademlia [221] provide $\mathcal{O}(log(n))$ complexity for both. This means that each cover message requires $log(n)$ messages to be sent. The messaging complexity can be reduced to $\mathcal{O}(1)$ by accepting $\mathcal{O}(\sqrt{n})$ storage per peer. We consider the overhead for DHT cover traffic too large. Centralizing the message transfer infrastructure improves performance and reduces load on end devices while providing the same privacy and security guarantees.

In this section, we present Ovid. To the knowledge of the authors, it is the first DCT system that combines blind signatures and *postbox channels* with cover traffic to ensure user privacy against the health authority at all times. The parameters used in the evaluation of the required cover traffic provide a balance between privacy and performance.

The main contributions of Ovid are:

- Applying blind signatures to verify the authenticity of an infection message while ensuring that the health authority does not learn which users interacted.
- A defense mechanism against flooding the system with malicious warnings.

- A scalable concept for cover traffic.
- An evaluation of the cover traffic and the signature handout performance.

Section 4.3.1 presents Ovid's system design as well as additional explanations for the permission token mechanisms, the format of infection messages, the postbox retrieval mechanism, and operational aspects. Then, the security and privacy are analyzed in Section 4.3.2. The proposed system is evaluated in Section 4.3.3.

4.3.1 *System Design*

Co-location detection in Ovid is the same as in CAUDHT. Similarly, when a user Alice is diagnosed, she creates encrypted messages to her contacts that are signed by the health authority using a blind signature scheme. However, Ovid's messages additionally contain a time of encounter and a message time. Messages are then placed in the corresponding postbox channel. Each postbox channel encompasses multiple postboxes. A recipient Bob downloads the content of the postbox channels corresponding to his past pseudonym. If a message is successfully decrypted and the signature is valid, Bob learns that he has been in contact with a diagnosed person and is at risk of being infected.

By aggregating multiple postboxes into channels, accessing a specific postbox does not leak one of Bob's pseudonyms. Additionally, neither the server nor an eavesdropper on the network is able to deduce Bob's infection status by checking if any messages are addressed to him. Messages in the same channel double as cover traffic for other postboxes. Compared to broadcast-based DCT systems, this approach requires less communication. Postbox channels consist of postboxes sharing the same prefix p, e. g., the first 20 bits of the pseudonym. Channels are hosted on a single server, but they can also be distributed between several hosts.

Infection Messages

A newly diagnosed user needs to spread the news quickly to all users they have come across while being contagious. Let us assume Alice encountered Bob in the past and recorded Bob's ephemeral pseudonym P_{Bob}. Several days later, she tests positive for Covid-19. To warn Bob, Alice creates an infection message. The pseudonym is the public part pk_{Bob} of an asymmetric key pair, which she uses to encrypt the epoch of the encounter t_e. The epoch is a global numeric value that increases whenever a new pseudonym is used. She appends the current time t_m. Alice then blinds this string to retrieve a blind signature σ_{HA} from the health authority. The permission token required to retrieve a signature was given to Alice by the healthcare provider she visited to get tested.

Alice appends the signature to the first part, constructing the infection message m. The format for m is as follows:

$$m = E_{pk_{Bob}}(t_e)||t_m||\sigma_{HA}(E_{pk_{Bob}}(t_e)||t_m)$$

Alice stores m in the postbox channel corresponding to pk_{Bob} (see Figure 4.1). The channel is given by $truncate(pk_{Bob}, p)$ [13]. Alice will warn all other users she encountered during the relevant time period using this pattern. Having t_m as part of the infection message gives the postbox server the ability to figure out how old a signature is. The postbox server does not accept infection messages with a timestamp t_m in the future and can remove those older than the maximum incubation time. Since the signature covers t_m, an attacker cannot overload the postbox system by replaying old infection messages.

[13] *The function* $truncate(x, p)$ *returns the first* p *bits of* x.

To hinder an eavesdropper located on the network from figuring out if Bob received an infection message, cover traffic is required. Aggregating postboxes into postbox channels gives Bob plausible deniability. However, this might not be enough if there are only few messages in the system. Therefore, when Alice receives her bulk of permission tokens from her healthcare provider which allows her to retrieve signatures, she might receive more than she asked for. The probability for additional tokens can be either fixed or dependent on the current utilization of the system. After obtaining signatures for all entries in her history of encounters, Alice creates random messages and fetches valid signatures for these by using her remaining permission tokens. These cover messages are addressed to random postboxes. When Alice places all her messages in the postbox system, the server will not be able to differentiate between real infection messages and signed cover traffic. As all unsigned messages are discarded by the server, this means only diagnosed users can create cover traffic.

Retrieving Blind Signatures Using Tokens

Before signing a blinded message $b(m)$, the health authority needs to verify the request originates from a user with a confirmed infection. People with symptoms of Covid-19 visit healthcare providers to get tested. In Ovid, permission tokens are passed from the healthcare provider to the newly diagnosed person when a test returns positive. These tokens allow the diagnosed user to retrieve blind signatures from the health authority for the messages they intend to upload. Each token authorizes one blind signature.

There are two ways in which these tokens can be designed. In one variant, the health authority could centrally generate random numbers to be used as tokens and distribute them to healthcare providers who conduct testing. Assuming the health authority is not compromised and the space from which tokes are drawn is big enough, it is not feasible for an attacker to generate fake tokens. A downside of this approach

is the possible linkability of a token to a specific location or healthcare provider when a patient uses it to request a blind signature.

A second option for token creation, which is used by Ovid, is that healthcare providers generate and sign tokens themselves using a *Public Key Infrastructure* (PKI) to distribute their public keys. This way, the health authority can ensure a token originated from a valid source. To stop the health authority from finding out which healthcare provider a user visited, we use a *ring signature* scheme [265]. With ring signatures, it can be verified that one of a predefined set of keys was used for signing, but not which specific one. This means the actual signer (here, a healthcare provider) is not linkable to the signature. The ring signature's size grows linearly with the number of signers in the ring [208]. It would become impractical to have one ring for all healthcare providers, both in terms of signature verification complexity and size of the ring's public key. We propose to form smaller rings consisting of several healthcare providers each to ensure a balance between signature size and anonymity set. This keeps key sizes manageable and allows for member changes in the system without having to discard the entire signature ring. To prevent a curious health authority from estimating the geo-location of a diagnosed patient presenting a token, each ring is filled randomly so that all ring members are distributed over the health authority's geographical region.

Ovid uses this ring signatures-based mechanism for token generation. Tokens are composed as follows. Each token consist of a unique 32 byte random number and a timestamp, as well as a signature. The timestamp has a granularity of 14 days and is used to prevent an attacker from replaying recorded tokens to generate additional blind signatures. A signature over the random number and the timestamp verifies the token's authenticity. The token is generated and signed by the healthcare provider who hands it out. After a token was used by the diagnosed user, it is placed on a blocklist maintained by the health authority. Entries on the blocklist can be discarded after 14 days have passed because of the timestamp in the tokens.

Postbox Retrieval

Users need to query their postboxes periodically if they want to determine whether they are at risk.

If Bob performs a search for the truncated pseudonym in the postbox service all messages from this postbox channel will be returned. He will attempt to decrypt all returned messages using the private key corresponding to P_{Bob} and succeed with Alice's message. To give Bob a fast way to check if the decryption of $E_{P_{Bob}}(t_e)$ was successful, a fixed amount of zeros can be added to the beginning of t_e before it is encrypted. This way Bob knows the message was addressed to him. The signature part of the message confirms to him that Alice's test result

was indeed positive. The decryption gives Bob the timestamp t_e of their encounter. He performs a sanity check to verify that P_{Bob} was advertised during that time. This stops an attacker who tries to create false warnings by collecting pseudonyms in a low-risk area and replaying them later in a high-risk area. Since no part of the message contains any information relating to Alice, Bob will not learn that the message was created by her.

Users are notified by their end device that they are at risk after a certain threshold of exposure to diagnosed users is exceeded. This threshold needs to be defined by epidemiologists. Since risk assessment is done locally, it is possible to take individual risk factors into account without endangering users' privacy. Such factors can be their medical history, adherence to mask mandates, or the general infection risk in their area.

Operation

To ensure that the system is operated in a secure and privacy-preserving manner, it is important to consider some implementation details. A theoretically secure system can be dangerous for users if the operator uses outdated cryptographic functions or if the end-user app contains security vulnerabilities.

Our first consideration is the correct key management and distribution. For the blind signatures to be verifiable and secure against manipulations from a network operator, a secure channel for the health authority's public key to the end user is required. The end user app needs to verify the health authority's key, which is why the health authority needs their key either signed by a widespread SSL certificate provider or directly embedded into the end user app. Theoretically, the connection to the health authority does not need to be secured itself because the blind signature scheme itself provides a verification mechanism for the receiver of the signature. To prevent a DoS attack by the network operator who could block packets with messages containing blind signature requests, it can still be beneficial to use an SSL-secured connection. To verify tokens, the health authority needs the public key for all rings of healthcare providers. This can be either done via direct communication or also using a PKI. On the client side, it is important to have a proper key store for managing the secret keys in the app. These keys should be stored in encrypted storage sections if the operating system offers such a service. Alternatively, the app could prompt the user with a password request to encrypt their secret keys. Such a prompt has to be carefully designed so that usability (and thus usage) is not harmed by a complex user interface.

In case of a high infection risk, users should be notified as fast as possible. This requires users to poll the postbox services in short intervals. To support this traffic overhead, a scalable infrastructure such as con-

tent delivery networks can be used to handle the load. When sending a request to retrieve a postbox channel, users can send the timestamp of their last request. The server can then return only messages that the user has not previously fetched. This decreases the network load for users and for the postbox service.

Users have to decrypt all messages they receive to find out if they are at risk. The decryption of messages in asymmetric encryption schemes is a CPU-intensive task and therefore requires more energy. For this reason, this task should be executed when the user's end device is charging or the battery is reasonably full. At least once per day, all new messages need to be decrypted to ensure timely notification.

4.3.2 *Security and Privacy Analysis*

To evaluate the privacy and security properties of Ovid, we discuss the most relevant attacks in the following paragraphs. All attacks on BLE proximity detection as presented in Section 3.6.1 are relevant for Ovid. The same goes for the attacks on deanonymizing diagnosed users from infection messages presented in Section 3.6.2. Additionally, the health authority and network eavesdroppers can analyze metadata as described in Section 3.6.3. Defense mechanisms for the individual attacks are presented in the corresponding sections.

A *false notification* or *black market attack* has the goal of sending infection messages to users who are not at risk. To do so, an attacker needs to obtain valid pseudonyms to send messages to. This can be done by recording advertisements or through a black market exchange. An attacker simply guessing pseudonyms is not considered a threat to Ovid because the space from which pseudonyms are drawn is too large. To execute the attack, the adversary needs a permission token to retrieve a valid signature from the health authority. They then place an infection message in the corresponding postbox. As long as the epoch of the encounter t_e matches the epoch in which the target used the corresponding pseudonym, the target will falsely assume that the message is valid. To mitigate false notification attacks, the message format could be extended to include a signature over the pseudonym of the diagnosed user used during the encounter.

A curious stalker Stan can capture pseudonyms of his target Tiffany and snoop on the corresponding postbox channels. Cover traffic stops him from being sure if there are real infection messages for Tiffany. For rates of cover traffic of at least 10% and more than two users per channel, this attack becomes inefficient (see Figure 4.2).

Defenses against metadata leakage are relevant not only when uploading data to the health authority but also when diagnosed users try to retrieve signatures. When checking postboxes, it is not necessary to use anonymization networks or similar methods because of the cover traffic. The health authority can also conduct a *timing attack*, as the

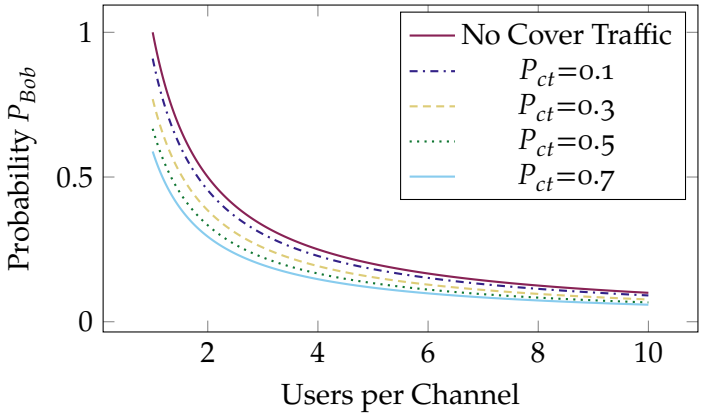


Figure 4.2: The probability of a message in a channel being designated for user Bob, who accesses the channel. It is plotted against the number of users per channel. Each curve illustrates a different probability P_{ct} representing the likelihood with which a user uploads an additional cover message for each real message. Figure as in [4].

sender will upload messages as soon as they are created. As mitigation, a diagnosed Alice can spread out her uploads over several hours. As this delays the time at which users at risk receive a risk notification, the uploads should not be spread out too far.

An eavesdropper only listing to network traffic cannot derive if messages received by a user are infection messages that contain a warning. This is because real messages for a specific user are hidden between real messages for other users and cover traffic.

4.3.3 *Performance Analysis*

To assess the practicality of Ovid, a performance evaluation is conducted using a Python REST server. The server provides blind signatures, stores infection messages uploaded by diagnosed users, and distributes them to requesting users. The server runs on a machine with three dedicated Intel® Xeon® E5-2643v2 cores and 8 GB of RAM. The client is capable of creating infection messages and querying the server for messages using its past pseudonyms.

First, we evaluate the performance needed by the health authority server. The server is capable of providing 4651.16 $\pm$ 259.55 blind signatures per second (10 runs on localhost). Here and in the following, error intervals represent the standard deviation [14]. Generating blind signatures can be parallelized by sharing the health authority's secret key with several physical machines. Even in scenarios where a single diagnosed user uploads hundreds of messages, the system remains scalable.

Our second focus lies on the performance of the backend database. To be able to easily discard entries that are older than 14 days, the

[14] *The standard deviation was selected as it represents a deviation from the mean, which is a useful property when analyzing expected runtimes.*

time-series database InfluxDB was used. Storing 10,000 messages in InfluxDB took $6.33 \pm 0.73\ s$ (10 runs on localhost).

To assess the performance of Ovid, it is relevant to understand the level of cover traffic required. This is influenced by the size of the postbox channels and the probability P_{ct} that a diagnosed user receives an additional permission token per requested token. These additional tokens are used to create random messages for cover traffic. The probability of how likely it is for a packet to be addressed to Bob is $\frac{1}{m}$ where m is the number of users per channel. The probability that a packet is not cover traffic is $1 - \frac{P_{ct}}{1+P_{ct}}$. Combining both gives the probability that a message in a postbox channel is actually for Bob as $P_{Bob} = \frac{1}{m} \cdot (1 - \frac{P_{ct}}{1+P_{ct}})$. See Figure 4.2 for visualization with different values of P_{ct}. We see that the influence of users per channel is stronger than the influence of P_{ct}.

In our third experiment, we assess Ovid's reporting performance over a 100 MBit fiber Internet connection in a large setup, with a delay of 50 ms. We simulate 3,000 clients on an Intel® Core® i7-8550U and 16 GB of RAM. Clients use encounter histories generated by a script. Each user is assigned a number. Encounters are created by drawing two numbers from a uniform distribution where each number represents a user. The corresponding epoch is derived similarly by drawing a discrete time interval from a uniform distribution. For each encounter, the corresponding history files are updated by appending the other side's public key from that epoch. Each history contains 100 encounters on average. During the simulation, clients create and upload infection messages for all of their generated encounters. P_{ct} is set to 0.1 as Figure 4.2 shows that higher levels of cover traffic do not come with additional privacy. We do not consider a probability of less than 0.1 for cover traffic, as the parameter ensures that some artificial messages are stored on the server. This is especially relevant in situations with low infection rates and thus few real messages. The average reporting time is $279.33 \pm 142.27\ s$ for a client over the previously described Internet connection. Users who do not create infection messages were not simulated as they only query the server but otherwise do not impact the system.

In our final experiment, we evaluate the retrieval and decryption performance of a client. Again, we simulate 3,000 clients as described in the previous experiment. We assume that a channel is identified by the first 19 bit of the recipient's public key, resulting in 2^{19} channels. This provides a relatively large anonymity set even in scenarios where infection rates are low, leading to a little more than 5 users per channel on average. As can be seen in Figure 4.2, such an anonymity level seems reasonable to us, as more users per channel only slightly improve anonymity but increases the load on the network and decryption times on the client. Users have 1008 pseudonyms, which corresponds to a new pseudonym every 20 min over the course of two weeks. This means a user has to query approximately 1008 postbox channels. The level of cover traffic P_{ct} is set to 0.1. A client retrieves all channels that

correspond to its pseudonyms and attempts to decrypt the messages contained in them. If a message is decrypted successfully, the risk score is updated. The average duration to retrieve and decrypt infection messages from postboxes covering the last 14 days was $79.80 \pm 27.83\ s$ (100 runs over the Internet). Although decrypting messages for a user is a resource-intensive task, performing it once a day appears to be feasible on mobile devices.

As we have seen in this evaluation, Ovid is scalable and provides adjustable privacy levels depending on the current pandemic situation.

4.4 CHAPTER SUMMARY

In this chapter, two approaches to DCT were presented that improve the proposal of Cho et al. [75] for client-side risk assessment through direct messaging. In the case of Ovid, a central health authority is only required to provide blind signatures and host the postbox system for delivering messages. It can not learn sensitive information from providing these services. For CAUDHT, the health authority is even less involved as messages are delivered via a distributed infrastructure hosted collectively by all users. Both, the centralized and the decentralized messaging service, require cover traffic to hide the fact that a user has received a warning. While using a DHT provides availability and scalability, the overhead for cover traffic is significantly larger despite providing the same privacy properties as a centralized service. We recognize that by merging postboxes into postbox channels, real messages can be used as a natural source for cover traffic. This ensures probabilistic protection against an honest-but-curious messaging delivery service, network eavesdroppers, and curious stalkers.

Another contribution is using blind signatures to provide authenticity. This ensures that the signing authority – here, the health authority – can not identify sender-receiver pairs.

Stronger privacy guarantees can be provided by leveraging cryptographic primitives in the system design. The following chapter will discuss two approaches that aim to leak as little data as possible to both the health authority and the users.

5

CRYPTOGRAPHIC APPROACHES TO DIGITAL CONTACT TRACING

The main goal of client-side Digital Contact Tracing (DCT) approaches, like those presented in the previous section, is to remove trust from authorities like the health authority or other governmental institutions. This aims to ensure the voluntary participation of people in contact tracing efforts.

However, client-side risk assessment uses Bluetooth Low Energy (BLE) for proximity detection and can not easily be adapted to other types of data. Cryptographic protocols, on the other hand, are very versatile while providing similar or even stronger privacy guarantees. This chapter examines how a cryptographic protocol can be used to allow DCT based on either location data or BLE-based proximity detection. Here, if GPS data is used to detect encounters this does not mean that location traces of diagnosed people have to be made public. Additionally, the location privacy of healthy users and those at risk is guaranteed.

Another advantage of cryptographic protocols is the fact that they can minimize the leakage of sensitive data to both client and server. An attack that is feasible against all DCT approaches with client-side risk assessment is the deanonymization of diagnosed users due to the leakage of the encounter time or epoch (see Section 3.6.2 of Chapter 3). Protocols like *Private Set Intersection* (PSI) can mitigate this leakage. In this chapter, we also show how PSI can be used to achieve risk-scoring functionalities on a par with those proposed for the broadcast-based systems Google Apple Exposure Notification (GAEN) and DP-3T while defending against this simple but effective attack.

This chapter is structured as follows. Section 5.1 explains several cryptographic protocols for privacy-preserving computation. Related work is presented in Section 5.2. Section 5.3 introduces an early idea from the beginning of the pandemic that relies on *Oblivious Random Access Memory* (ORAM) techniques for DCT on location and BLE data. Last, our proposal CERTAIN is presented in Section 5.4. This DCT system leverages circuit-based PSI to implement complex risk-scoring functionalities. Here, users only learn their final risk score and nothing else, which removes the attack surface for the deanonymization attack described above.

5.1 ON PRIVACY-PRESERVING COMPUTATION

Cryptographic protocols for privacy-preserving computation allow multiple parties to jointly compute a result from private inputs without revealing the data to one another.

Homomorphic encryption [291] encompasses a set of encryption schemes that allow computation on already encrypted data. A homomorphic function is defined as follows: Let $f(x_1, x_2, \dots, x_n)$ be a function with n inputs and let function h be the corresponding homomorphic encryption function. For an encryption function $e(x)$ and the corresponding decryption function $d(x)$ it holds that:

$$d(h(e(x_1), e(x_2), \dots, e(x_n))) = f(x_1, x_2, \dots, x_n)$$

The decrypted result will contain the same result as if f was applied to unencrypted data. The most advanced types of homomorphic encryption are *fully homomorphic* schemes which allow an unlimited number of multiplication and additions on encrypted data.

The runtime is a great weakness of homomorphic encryption schemes and depends on the multiplicative depth of the function to be computed. As a result, some operations are significantly faster and easier to implement than others.

Multi-Party Computation (MPC) [291, Chapter 22] facilitates joint computation on private, distributed data. It studies mechanisms to allow a group of n independent participants to collectively evaluate a function $y_1, \dots, y_n = f(x_1, \dots, x_n)$. Each participant i holds a secret input x_i, which remains hidden from other parties but is used for computation. The participants only learn their designated final result y_i. Any function that can be mapped to a finite-sized circuit can be computed with MPC. This includes all functions that are computable in polynomial time [291, Chapter 22.2].

One way to implement MPC protocols is *Yao's garbled circuits* [116, 291]. Standard garbled circuits are only applicable to the case with two parties, a garbler and an evaluator. Here, one participant creates a digital circuit for the function to be calculated and sends it to the other participant who evaluates the circuit. Evaluation requires oblivious communication between the evaluator and the garbler. Garbled circuits require a fixed number of communication rounds at the start of the protocol. Garbled circuits use boolean functions. However, it is also possible to compute arithmetic circuits, which are especially relevant for machine learning applications. The popular *Goldreich-Micali-Wigderson* (GMW) protocol [116] can compute both types of circuits. It can easily be applied to the case with multiple parties. Here, all parties simultaneously compute on secret-shared data. For each AND gate, a communication round is required.

General purpose MPC protocols can be slow, as they do not allow to make shortcuts or use domain knowledge. However, MPC protocols

can also be crafted to implement a specific functionality. Here, efficiency in communication and computation can be gained, e.g., by selecting the fastest protocol for a specific sub-problem and using general-purpose MPC only when necessary. *Private Set Intersection* (PSI) is an example where dedicated protocols are faster than general MPC protocols. Here, two parties want to find the intersection of their two data sets without revealing elements that are not in the intersection. A popular PSI protocol is *Diffie-Hellman PSI*. Here, both client and server first need to create an asymmetric RSA key pair. Each side encrypts their set with their private key and sends it to the other party. The recipient then encrypts the already encrypted set with their key, so now each set is encrypted with both private keys. The server sends the set it encrypted last to the client, which then holds both sets. The client calculates the intersection of these encrypted sets. Due to the multiplicative property of asymmetric encryption, it is not important which key was used first.

5.2 RELATED WORK

As mentioned in the systematization in Section 3.4.1 of Chapter 3, there are various approaches using homomorphic encryption, MPC, and PSI for contact tracing. To compare these against the two designs proposed in this chapter, additional detail is provided for all the designs that do server-based risk scoring without revealing the results to the server [15]. Details for client-side risk-scoring approaches have been discussed in Section 4.1. Unlike cryptographic protocols, they are less flexible and only allow BLE pseudonyms to be used for tracing. However, performance is better.

[15] *See Table A.1 in Appendix A for a tabular overview of all the mentioned approaches.*

EPIC

The Epic Framework [29] from 2018 relies on homomorphic encryption for DCT. Similarly to the first variant of the approach presented in Section 5.3, it relies on location-based data. However, location are fingerprinted with Wifi and Bluetooth. No active Bluetooth signaling is conducted. Such location fingerprints captured by diagnosed users are uploaded in plain text to servers belonging to the health authority. Undiagnosed users send requests to the server to determine how similar their location fingerprints are to those measured by diagnosed users for certain timestamps. The request contains the public key of the user, the timestamp t_e, and an (homomorphic) encryption of the location fingerprint at t_e. The server will use the provided public key to encrypt location fingerprints with a close timestamp and then calculate a matching score. The scores cannot be decrypted by the server. It will send the result back to the requesting user who can decrypt it and derive their personal risk score. Users do not learn the location traces of individual diagnosed users but will learn at which locations they have

been close to a diagnosed person. They can also forge their upload to verify assumptions about the risk status of a person. Service operators are not able to learn the locations, risk scores, or health status of healthy users because data is encrypted with a secure key belonging to the user.

TraceSecure with Homomorphic Encryption

Another approach using homomorphic encryption was proposed by Bell et al. [46]. The system relies on pseudonyms exchanged over BLE. It reveals to the server (which is run by the health authority or a service operator) who has interacted with whom but keeps the health status secret from the server and non-colluding network operators. This leakage of interactions can be used for building a social graph of pseudonyms. Bell et al. consider this graph to be a feature as it can be used as part of a privacy-preserving evaluation of social distancing policies. Users of homomorphic encryption-based TraceSecure learn which pseudonym was responsible for a warning.

Demirag et al.

Multiple DCT designs that use cryptographic protocols rely on PSI. The protocol of Demirag et al. [92] is one such example. Here, standard proximity detection via BLE advertisements is considered. The health authority server holds all pseudonyms of people with verified infections. To figure out how many users they have met in the last weeks that were diagnosed, a user performs PSI with the server following the protocol of De Cristofaro et al. [86]. This protocol only returns the size of the intersection. This means no complex risk scoring like in CETRAIN is possible. The system requires the central server to know relevant information about the diagnosed individuals, here the pseudonyms they have used in the past. The server (i. e., the health authority or a service operator) does not learn which pseudonyms the client used as input for the set intersection or if they are at risk. The client does not learn the pseudonyms of users that are diagnosed, not even the ones they have been in contact with.

Epione

Epione proposed by Trieu et al. [304] also uses Bluetooth technology to exchange pseudonyms. For each encounter, both parties create a new pseudonym. This design use a Diffie-Hellman-based PSI algorithm to determine the cardinality of the intersection. Their algorithm is optimized for situations with unbalanced sets, so where the client's set is considerably smaller than the server's set. This approach also uses homomorphic encryption for some steps. In Epione, the health authority and the central server are required to know the pseudonyms diagnosed

users have used in the past. The server does not learn the past pseudonyms of other users or who is at risk. The client only learns how many risky encounters they have had but not the corresponding pseudonyms of diagnosed users. This means that no complex risk scoring is possible as done by CERTAIN.

Berke et al.

Like Epione, Berke et al. [49] also uses Diffie-Hellman PSI. However, it computes the overlap of GPS traces from diagnosed people with traces provided by individual users. Coordinates are truncated and rounded so that they are represented by single dots on a three-dimensional grid (longitude, latitude, and time). Since distance is an important factor when transmitting the virus, for each truncated coordinate it is important to check whether the neighboring grid points are part of the intersection. The PSI protocol can be used to allow clients to learn the size of the intersection, but also which of their elements appear on the servers by letting the client query for elements individually. The server, and therefore the health authority or a service operator, does not learn which data was provided by a user. Users do not learn the location history of diagnosed people they have not met. The client learns if they are at risk and since the intersection is leaked they will also know where the encounter occurred. Since GPS data is used, malicious users can forge input and for example provide the home address of a target.

The functionality of this protocol is similar to our ORAM approach presented in Section 5.3. However, as each GPS location is queried individually instead of comparing sets, the complexities differ. By using a circuit-based PSI protocol as proposed by our second approach CERTAIN in Section 5.4, the location privacy of diagnosed users could be improved.

5.3 PRIVACY-PRESERVING CONTACT TRACING USING OBLIVIOUS RANDOM ACCESS MEMORY

This section is based on previous work with Samuel Brack and Björn Scheuermann presented at the poster session of the IEEE Symposium on Security and Privacy (S&P) in 2020 [5].

In the wake of the pandemic, Israel conducted contact tracing on location data derived from cell tower triangulation to stop the spread of Covid-19 [31, 147]. While effective at identifying infected people during the early phase with a low incidence rate, the cuts to citizens' rights due to such mass surveillance measures were severe. This section proposes a solution for preserving the location privacy of undiagnosed people in such a centralized contact tracing system. Here, either location traces or BLE pseudonyms of diagnosed users are stored in an ORAM. Users can query the ORAM to determine their current risk of infection.

ORAM and the PathORAM protocol are introduced in the following. Section 5.3.2 presents the system design. The performance of this ap-

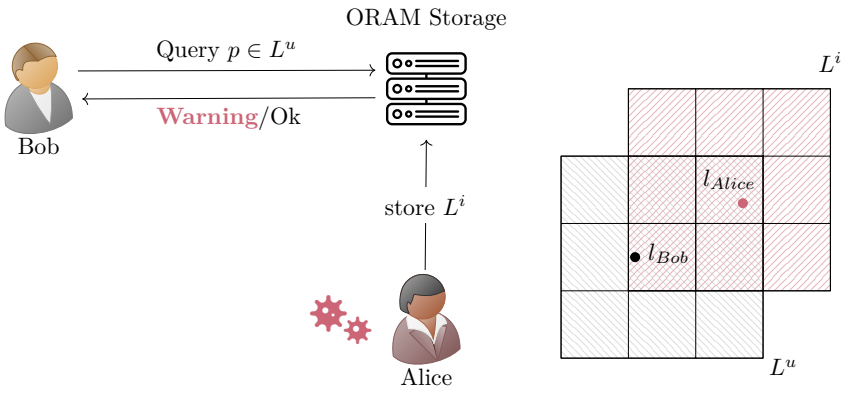


Figure 5.1: An infected person Alice shares her location data with the health authority. Bob queries all his past locations to find out if he recently crossed paths with an infected person. Locations have multiple dimensions. A data point l is rounded to the closest position on the grid. Using this as the center, the set L of adjacent grid locations is computed, covering the region close to l. If a set L^u belonging to a Bob intersects with the set L^i of an infected individual, then Bob is at risk of having contracted the disease.

proach is analyzed in Section 5.3.3. Section 5.3.4 discussed aspects of privacy, security, and utility.

5.3.1 *Oblivious Random Access Memory*

Oblivious Random Access Memory (ORAM) [104] is a MPC protocol for privately storing and reading encrypted data from a semi-honest server. It hides what data was requested by a client by obfuscating the accessed index i. The ORAM ensures that a sequence of read or write operations are indistinguishable from one another.

PathORAM [292] is a type of ORAM protocol that gained popularity due to its excellent performance. In PathORAM, the server organizes blocks that contain data into buckets. These buckets are arranged in a binary tree T_{ORAM}. Each bucket can fit Z blocks. The client maintains two additional data structures: the position map and the stash. The position map associates the IDs of blocks to leaves in T_{ORAM}. When retrieving the path to a leaf L_i, block B_i is located in one of the buckets on this path. The stash is a temporary data structure that stores blocks that have been retrieved from the server. The complexity of a read or write operation for PathORAM is $\mathcal{O}(log(N))$ where N is the maximum capacity.

Standard PathORAM requires the client to maintain certain data structures. To achieve multi-user access, these can be outsourced, for example, to a Trusted Execution Environment (TEE). This section as-

sumes the existence of a multi-user PathORAM. Chapter 8 goes into detail about how PathOram needs to be adapted for TEEs.

5.3.2 *System Design*

During manual contact tracing, health authorities collect location histories of infected users (see Section 3.1). Our proposed system (see Figure 5.1) takes advantage of this fact and aims to preserve the location privacy of all others. We assume that a vast majority of individuals use location-based services that record their movement, such as the Google Location History [138].

Input locations $l := (x, y, t)$ consist of geographical coordinates and a temporal component. The altitude is ignored as it is deemed too error-prone. Each user u has m locations in their location history for which they want to check if they have come in contact with an infected person there. The health authority holds the location history of all recently diagnosed individuals which consists of n data points in total. The location history of infected users only needs to contain data points from the duration when they were infectious.

For a location l each component is rounded to a fixed granularity (e.g. 1 meter or 1 minute) so that it can be represented by a position on a multi-dimensional grid. The selected granularity only depends on epidemiological factors. For each l, a set L of locations on the grid is calculated for which the Euclidean distance is smaller than a fixed threshold. See the left side of Figure 5.1 for a visual representation of L for a single location. Due to the reduced granularity, the number of elements in L is small.

Both the health authority and the user compute L for all their respective data points. The health authority stores its results in an ORAM. It can either host the ORAM itself or ask a third party to provide this service. An ORAM allows data to be read from a remote server without revealing to the server what entries were accessed.

For each location and each element in the corresponding set L^u, the user initiates a search on the ORAM. If an element from L^u is found in the ORAM, then the region described by L^u intersects with the set L^i representing a location visited by a diagnosed user. This means the user has been in contact with a diagnosed individual. The number of contacts and their duration can be used to derive a risk score on the client's side. Due to the usage of an ORAM as storage for location data, the server of the health authority will not learn what locations were queried by users and only the users themselves will learn the result of their query.

The algorithm described above is guaranteed not to leak more information than the ideal functionality to either party's side. For threat analysis purposes, participants of the system are modeled as semi-honest.

Such a model can be reinforced to provide security in a malicious setting by accepting a performance penalty [134].

5.3.3 *Performance Considerations*

The evaluation from the original paper [5] was improved and extended for this thesis.

We used the PathORAM implementation of Epsolute [55] to test the presented approach.

Due to availability, the measurements were done on an AWS server with 121 GB RAM and 16 AMD EPYC 7013 processors which provides the TEE platform AMD SEV-SNP. The resources were dimensioned larger than necessary for the measurements and only a small portion was used. Figure 5.2 shows the runtime for a single ORAM read for different ORAM sizes. As we can see, the runtime for one read increases logarithmically with the size of the ORAM and takes less than 0.8 ms even for 2^{30} data points.

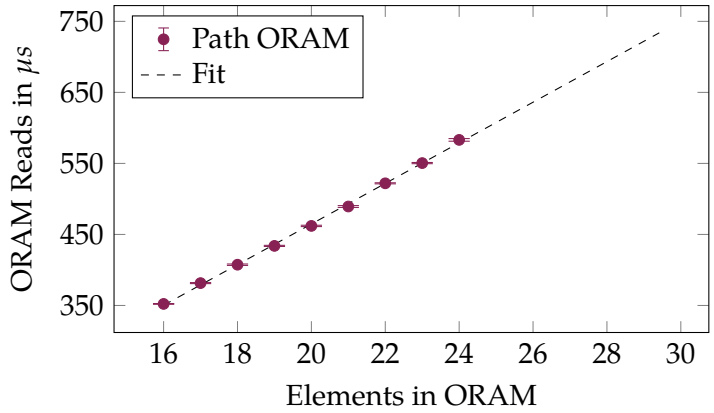


Figure 5.2: Runtime measurements for reading a single element from a PathORAM. The error bars (in gray) show the 95% confidence interval. Linear regression on powers of two was used for fitting.

The size of the ORAM required for DCT depends on various parameters. To estimate the runtime, let us assume data is recorded in one minute intervals for 16 h a day, accounting for 8 h of sleep. Let the blowup for a single location l be $|L| = 9$. This means that all adjacent grid locations, except in the time domain, are queried to check a single location. It follows that one diagnosed person uploads 960 data points for each day. Data points older than 14 days do not need to be uploaded or can be removed as they do not contain relevant epidemiological information anymore. The number of data points stored in the ORAM differs depending on the daily number of newly diagnosed people. While retrieval runtime for one ORAM entry is reasonably short, it is essential to remember that each user has to check all locations recorded during the last 14 days to identify encounters with diagnosed people.

This process requires $960 \cdot 14 = 13,440$ read operations per user per day. See Table 5.1 for an overview of the corresponding runtime.

Table 5.1: Runtime comparison for using location data and BLE ephemeral pseudonyms for contact tracing with a PathORAM.

	Location Data		BLE Data	
Newly Diagnosed People/Day	Runtime per User	ORAM Size	Runtime per User	ORAM Size
10	5.47 s	2^{18}	0.66 s	2^{21}
100	6.57 s	2^{21}	0.77 s	2^{24}
1,000	7.86 s	2^{24}	0.89 s	2^{27}
5,000	8.93 s	2^{27}	0.96 s	2^{29}
Uploaded Location per Diagnosed Person	13,440 Locations		140,000 Locations	
Reads per User/Day	13,440 Reads		1,344 Reads	

Let us look at an alternative setting where BLE beacons are used for proximity detection instead of location traces. Let a new ephemeral pseudonym be generated every 10 min. To keep the number of ORAM reads low and reduce the load on clients, diagnosed users upload the pseudonyms of those they have been in contact with. This means each user trying to determine their risk only needs to query 1,334 entries in the ORAM daily. Using the assumptions of the DP-3T authors, each diagnosed user uploads about 140,000 different collected pseudonyms from the last 14 days when receiving their diagnosis [305]. It follows that the ORAM has to hold more data for BLE pseudonym contact tracing as compared to the location-based approach. However, as less data has to be queried per user, runtime improves by a magnitude.

5.3.4 *Discussion*

The proposed system uses a central party (the health authority) for DCT. Each person wishing to check their history for contacts with diagnosed individuals has to go through this central instance. Due to the privacy properties of ORAM, no sensitive data of querying users is leaked to other users or to the health authority. This stops a semi-honest user from learning the private data of diagnosed individuals. However, an issue with using location data for contact tracing is that a malicious user can iterate over potential locations, thereby extracting the location history of diagnosed people. As shown by privacy research [91], but also in the context of deanonymization attacks on Covid-19 infected people in South Korea [174], the location history contains information that allows for identifying the corresponding person. Here, circuit-based PSI protocols, as utilized by our proposal CERTAIN in the following section, can provide location privacy for diagnosed users.

Another attack vector to the proposed system is the possibility for a malicious health authority to create panic by adding additional fake locations to the ORAM. However, this attack would not help reach the overall goal of tackling a pandemic situation.

As discussed in Section 3.3 of Chapter 3, an issue of using location data is the high false positive rate [204]. Location data is inaccurate in indoor settings and proximity might be detected even if both sides are in different rooms. This and other factors are why BLE has been more popular for contact tracing. As shown above, an our proposed ORAM DCT system can handle both location data as well as BLE ephemeral pseudonyms to detect proximity. The latter even improves runtime by an order of magnitude.

Our main contribution lies in the application of MPC to the real-world problem of centralized contact tracing. On one hand, using MPC results in a significantly longer runtime than other centralized approaches. On the other hand, it provides semi-honest security, while a majority of centralized schemes rely on a trusted server.

5.4 CIRCUIT-BASED PSI FOR COVID-19 RISK SCORING

This section is based on previous work with Marcel Pazelt and Björn Scheuermann presented at the IEEE International Performance, Computing, and Communications Conference (IPCCC) in 2021 [7].

Geneal purpose MPC protocols allow to limit the amount of data learned by either side of a computation to the designated final result. This property can be leveraged for to defend against a simple but effective type of deanonymization attack against diagnosed users. Client-based contact tracing systems like GAEN, where pseudonyms of diagnosed users are published and risk scoring is conducted locally, suffer from the fact that a malicious user who received a warning can deanonymize the corresponding diagnosed individual by remembering who they have been in contact with during the time of the encounter (see Section 3.6.2). This attack is also feasible against CAUDHT in Section 4.2, Ovid in Section 4.3, and the system presented in Section 5.3 which relies on ORAM. PSI protocols that only reveal the cardinality of the intersection between pseudonyms of diagnosed users and the set of pseudonyms provided by the querying user, such as Epione [304], partially defend against this attack. However, a malicious user can strategically forge its input data set to derive the intersection. Additionally, a cardinality is not useful to derive a meaningful risk score based on exposure duration and distance.

To fill this gap, this section presents our system with complex risk scoring called CERTAIN. CERTAIN uses a circuit-based PSI protocol of Pinkas et al. [252] called OPPRF-PSI. Our contributions are:

- Designing an approach to using PSI that does not leak timing information.
- Devising complex risk scoring for using circuit-PSI with payload.

- Implementing an extension for OPPRF-PSI, which allows the inclusion of payload from client and server.
- Evaluating OPPRF-PSI for unbalanced sets with parameters aligned to the context of DCT.
- Examining the performance of the presented approach using an Android app.

We test a variety of circuits that align with existing risk-scoring functionalities. The system was evaluated with regard to communication, runtime, and energy efficiency in the context of for different networks.

In the following, the OPPRF-PSI protocol by Pinkas et al. [252] is explained in detail. Then, the design of CERTAIN is presented in Section 5.4.2. Section 5.4.3 focuses on the implementation and evaluations of CERTAIN and Section 5.4.6 discusses the privacy of the protocol. Other practical aspects of with CERTAIN are addressed in Section 5.4.7.

5.4.1 *OPPRF PSI*

Unlike non-circuit PSI protocols, which are often more efficient, OPPRF-PSI allows the computation of arbitrary functions on the output of the intersection without revealing intermediate results and additional information. Compared to other circuit-based PSI protocols like [76, 251, 253], it is the first to provide linear circuit complexity and a runtime $\mathcal{O}(n)$. This section explains a simple variant of OPPRF-PSI protocol without payload inclusion.

OPPRF

The functionality of an *Oblivious Programmable Pseudo-Random Functions* (OPPRF) can be described as follows. On a certain "programmed" set of inputs P the OPPRF outputs "programmed" values T, where $|P| = |T|$. When constructing the OPPRF, a *hint* is generated in the form of a polynomial. The hint is generated from a random key k, inputs P, and target values T. The polynomial maps each programmed input value $p \in P$ to the XOR combination of $F_{OPRF}(k, p)$ and its target value t. So, $F_{OPRF}(k, p) = t \oplus p$ [16]. To answer a query q, the hint is evaluated. If the OPPRF was programmed at position p and $p = q$ then the query returns the corresponding target value t. The OPPRF can be securely implemented as a two-party MPC protocol. Multiple OPPRFs can be batched to improve performance. This means that for each batch an independent OPPRF is executed.

[16] *Operator $\oplus$ represents a bitwise XOR.*

Simple PSI

Let the server set be Y and the client set be X. OPPRF-PSI computes the intersection of both sets as follows. First, the server uses simple hashing

to place its elements in β bins. The client uses cuckoo hashing [245] with the same hash functions on its set to create a distinct match from elements to bins [17]. The number of bins β is selected so that no stash is required during cuckoo hashing.

The protocol uses a batched OPPRF sub-protocol to determine if a client's element occurs in the corresponding server-side bin. For this purpose, it samples a set of target values $\vec{t} = \{t_1, \cdots, t_\beta\}$ from a random distribution. It then filles buckets $\{T_1, \cdots, T_\beta\}$ each with the respective value from $\vec{t}$ so that $|T_i| = |Y_i|$. Both sides then invoke β OPPRFs using the hashing bins of both sides and the server's target values. This protocol phase returns a vector $\vec{r}$ of size β to the client, which has the following property. If the client's element x_j (so the element that hashed to bin j) is contained in the server's bin Y_j, the value of the output vector $\vec{r}$ at position j is equal to the server's target value t_j. Otherwise, r_j is zero. Since communication is masked, the client cannot tell target values from zeros. In the next step, both sides compute a circuit with an MPC protocol providing X, $\vec{r}$, and $\vec{t}$ as input. The circuit compares for each position j if $\vec{r}_j$ is equal to $\vec{t}_j$ by using an AND gate. For this purpose, only γ bits are required to ensure fast runtime and a low false positive rate. The desired function f (e.g., a risk-scoring function) is then computed obliviously by using all elements from X for which this test was successful. The result is then revealed to the client.

The complexity of interpolating a polynomial grows at least linear in the number of encoded elements. To reduce computation, the polynomial can be split into a set of polynomials with lower size. To translate this optimization to protocol described above, bins are arranged in "mega-bins" of size β_m with a maximum of m elements in total. For each mega-bin, a batch-OPPRF is invoked.

[17] *Cuckoo hashing with one table and multiple hash functions: When a collision occurs on insertion, the original element is replaced and reinserted using another hash function.*

Payload Inclusion

Payload data is private data associated with an element used by either side for calculating the intersection. Payload is used to compute functions on the intersection. To be able to include payload data, some additional changes are needed to the basic OPPRF-PSI protocol. For payload inputs from the client, the adjustment is straightforward. The circuit has to be extended with input wires where the client inputs each element's payload. The result of $(\vec{r} \wedge \vec{t})$ is combined by an additional AND gate with the client's payload. More work is required when payload data originates from the server. Pinkas et al. [252] explain the problem and give instructions on how to extend the basic protocol to allow the server to input payload data. They did not implement or evaluate this part themselves. See Figure 5.3 for an overview of the PSI protocol with payload inclusion. For simplicity, mega-bins are not considered.

In the basic protocol without any payload, the server maps multiple elements to each bin and the OPPRF assigns the same target value to all elements within a bin. If the circuit now detects a match between two bins of client and server it is impossible to infer which of the elements in the server's bin had matched. To make this possible, the protocol requires two invocations of the batch-OPPRF. The first invocation is the same as in the original protocol. The second invocation is for identifying which payload of the elements that the server had mapped to a bin is related to the match with the client's element. As before, the client has an input set X and the server has an input set Y. Let $U(x)$ and $V(y)$ denote the payloads associated with $x \in X$ and $y \in Y$ respectively. Let x_j be the client's element at position j in $\vec{x}$ (so after Cuckoo hashing) and Y_j be the server's bin j.

For the second OPPRF, the server sets the values associated with one bin j so that they differ depending on the index. More precisely, the server samples $\vec{\tilde{t}} = \{\tilde{t}_1, \cdots, \tilde{t}_\beta\}$ uniformly and computes the sets $\{\tilde{T}_1, \cdots, \tilde{T}_\beta\}$ where $\tilde{T}_j(i) = \tilde{t}_j \oplus V(Y_j(i))$ for $i \in \{1, \cdots, |Y_j|\}$ and $j \in \{1, \cdots, \beta\}$. It inputs the sets $\{\tilde{T}_1, \cdots, \tilde{T}_\beta\}$ into the OPPRF functionality. The client inputs its vector $\vec{x}$, like in the first OPPRF. The batch-OPPRF outputs the result vector $\vec{\tilde{r}}$ to the client.

Now, the circuit computes for index j the following:

1. The client inputs $x_j \in \vec{x}, r_j \in \vec{r}, \tilde{r}_j \in \vec{\tilde{r}}$ and $U(x_j)$. The server inputs $t_j \in \vec{t}$ and $\tilde{t}_j \in \vec{\tilde{t}}$.

2. The circuit compares r_j to t_j. If they are equal, the server's payload has to be reconstructed. If the element x_j is the ith item in the server's bin Y_j, then the value received by the client is $\tilde{r}_j$ where $\tilde{r}_j = \tilde{t}_j \oplus V(Y_j(i))$. Thus, the server's payload for bin j is $V(Y_j(i)) = \tilde{r}_j \oplus \tilde{t}_j$.

3. Next, a sub-circuit computes the desired function f on x_j, $U(x_j)$ and $V(Y_j(i))$.

This adaption of the basic protocol for payload inclusion results in the same asymptotic complexity. The circuit now handles payloads and computes the same number of comparisons as the basic circuit [252]. The actual duration of the OPPRF phase doubles in length since it is invoked twice for the same number of bins.

5.4.2 *CERTAIN*

The following section explains how CERTAIN uses circuit-based PSI to implement complex risk scoring for DCT.

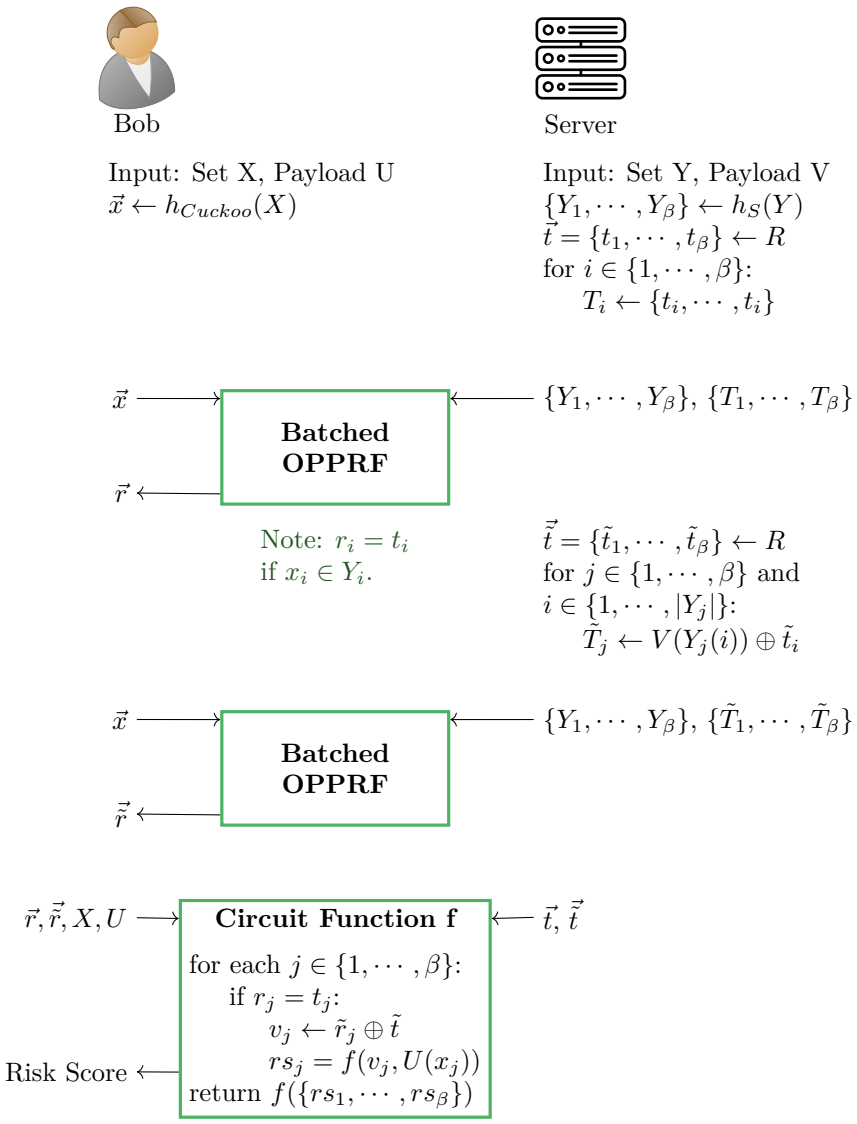


Figure 5.3: A simplified overview of the PSI protocol with risk scoring as used by CERTAIN. Both sides provide payload, which is used by the risk-scoring function f. Let V(e), respectively U(e), represent the payload associated with an element e.

System Overview

To participate in with CERTAIN, users download an app to their smartphone that regularly emits BLE advertisements. These advertisements contain ephemeral pseudonyms that change regularly. The app also collects pseudonyms of other users in the vicinity. To determine the distance to a sender, the signal strength is recorded. If a user is diagnosed, they pass the pseudonyms they used during the last 14 days to the health authority.

Users who want to determine their infection risk initiate a computation with the health authority's server to calculate a risk score using OPPRF-PSI. Each of the last 14 days is computed with a separated PSI run. The user inputs all pseudonyms recorded on the specific day, while the server inputs all pseudonyms of diagnosed users that were in use. As a result of the computation, the user will receive a set of risk scores. If the calculated risk for a day does not exceed a certain threshold, no risk is indicated. Otherwise, the risk score is revealed. Users can use these scores to decide whether to follow the health authority's directions and get tested. Since we assume this system is voluntary, reporting users who are at risk of being diagnosed is deemed counterproductive. Therefore, while it is possible, risk scores are not revealed to the health authority.

Risk Scoring Circuits

Risk scoring is the calculation of an exposure score, which reflects the risk of infection based on encounters with diagnosed individuals. The first version of the GAEN API risk-scoring approach [35] multiplies risk values for infectiousness r_i, duration r_d, days since exposure r_D and attenuation r_a per pseudonym e from the set of pseudonyms of diagnosed users D.

$$RS_{v1}^{GAEN} = \sum_{e \in D} r_{e,i} \cdot r_{e,d} \cdot r_{e,D} \cdot r_{e,a}$$

In the second version, changes in distance between users were also considered [35]. Here, duration at an attenuation range $j \in AR$ is multiplied with a corresponding weight w_j. The sum over all ranges is multiplied with a weight representing the infectiousness of the contact r_i and a value representing the reliability of the testing method r_{test}. GAEN defines four different attenuation ranges for immediate, near, medium, and other encounters. GAEN leaves the task of defining exact decibel values to the developers who build upon its API.

$$RS_{v2}^{GAEN} = \sum_{e \in D} \left(\sum_{j \in AR} w_j \cdot r_{e,j,d} \right) \cdot r_{e,i} \cdot r_{e,test}$$

Similarly, the mechanism of DP-3T [305] multiplies exposure at three different attenuation ranges with static weights and then calculates a

sum to determine the user's risk. The attenuation ranges are given by the thresholds 50 dB and 55 dB.

$$RS^{DP3T} = \sum_{e \in D} \left(\sum_{j \in AR} w_j \cdot r_{e,j,d} \right)$$

These examples show that both summation and multiplication are relevant for complex risk scoring.

We evaluate several functionalities for risk scoring using OPPRF-PSI. The most straightforward mechanism calculates the sum of payload values (`S`) provided by the client (`A`), which belong to an element that appeared in the intersection. We call this functionality `AS`. The payload can be, for example, the number of minutes $r_{e,d}$ the user was exposed to another person with pseudonym e. The circuit would then calculate the number of minutes for a day that the user was in contact with diagnosed individuals.

The next step is to allow both sides, client (`A`) and server (`B`), to provide a payload used for summation (`S`). This functionality is called `ABS` in the following.

Our complex risk-scoring functionality allows multiplying payload values (`M`) from both sides belonging to the same intersection element and then calculates the sum over all partial results. We refer to this functionality as `ABM`. It allows the health authority to provide information about the infectiousness for a specific pseudonym. If the client uses $payload_A(e) = r_{e,d} \cdot r_{e,a}$ and the server inputs $payload_B(e) = r_{e,i} \cdot r_{e,D}$ as payload for each element e of their sets, a scoring model similar to the GAEN API v1 can be achieved. To produce full GAEN or DP-3T risk scoring, which takes into account different attenuation levels, extra work by the client is required. For each recorded pseudonym e, it has to compute the following sum:

$$payload_A(e) = \sum_{j \in AR} w_j \cdot r_{e,j,d} \tag{5.1}$$

The overhead for this computation is minimal, as it can be calculated 15 min after the pseudonym was first received. For DP-3T risk scoring, the server does not need to include any data. For GAEN v2 the server has to include $payload_B(e) = r_{e,i} \cdot r_{e,test}$.

We additionally evaluate a set of circuits where a risk score is only revealed to the user if it exceeds a certain threshold (`T`). We apply this functionality to the three circuits described above, giving us the circuits `AST`, `ABST`, and `ABMT`.

Complexity for Unbalance Sets

The OPPRF-PSI protocol has a linear asymptotic communication overhead in the number of elements. However, the protocol is designed for intersecting sets of the same size. The effect unbalanced sets have on

the protocol complexity has not been discussed by Pinkas et al. [252], but is especially relevant in the case of DCT. This section takes a look at how the complexity of the overall protocol changes for the unbalanced case.

Each protocol phase is affected by this imbalance. Let n_1 be the client's set size and n_2 be the server's set size. Hashing takes $\mathcal{O}(n_1)$ for the client, respectively $\mathcal{O}(n_2)$ for the server. The complexity of set intersection is $\mathcal{O}(l \cdot n_1)$ gates where l is the bit-length of input elements. For circuits where both sides provide payload, the complexity is at $\mathcal{O}(l \cdot \beta \cdot n_1 + n_2 \cdot \delta)$ gates as payload inputs from the server with bit-length δ are included.

However, intersecting unbalanced stets influences the number of bins and mega-bins, which have an impact on the runtime complexity. Here, certain restrictions on bit length and failure probability become relevant. Pinkas et al. use Lagrange interpolation in a prime field based on the Mersenne prime $2^{61} - 1$. This allows for operations like multiplication of field elements to be an order of magnitude faster, but it limits the bit length γ of points to $\leq$ 61 bit. However, γ also depends on the failure probability of the PSI protocol (usually, 2^{-40}) and, therefore, the number of bins and mega-bins. The protocol fails if cuckoo hashing fails, if a collision in OPPRF outputs occurs due to an insufficient bit length γ, or if elements within a bin or mega-bin collide due to insufficient bit length l of the input elements. The 61 bit Mersenne prime field results in a maximum number of 1024 elements per OPPRF, i.e., per mega-bin. This means the number of mega-bins grows with the server's set size n_2.

When a total of β mega-bins is reached, the total number of bins β needs to be increased to ensure that no more than 1024 elements exist in a mega-bin. Per default, β only depends on n_1. To account for larger server set sizes, it is scaled by a factor ρ. Increasing β influences other parts of the protocol, such as the circuit size for computing the function f. As a result, the complexity of set intersection without payload is $O(l \cdot \rho \cdot \beta \cdot n_1)$, respectively $O(l \cdot \rho \cdot \beta \cdot n_1 + n_2 \cdot \delta)$ gates with server payload.

For the unbalanced case, the data required for hint communication is in $O(n_2)$ and the basic circuit is in $O(n_1)$. Therefore, performance has to be re-evaluated. Another factor influencing the protocol complexity is the fact that adding server payload for risk score computation doubles the OPPRF phase.

5.4.3 *Evaluation*

The key goal of this evaluation is to collect data from experiments to find out how well CERTAIN performs. This is primarily a question of efficiency.

5.4.4 *Implementation*

To build CERTAIN, we used a re-implementation of the OPPRF-PSI protocol [112]. It relies on the ABY framework [93] for circuit implementation. Using the Android Native Development Kit [32] the OPPRF-PSI code was ported to Android by cross-compiling all dependencies. We improved the protocol to allow payload inclusion and implemented an Android app to conduct experiments.

The OPPRF-PSI library code, as well as some of its dependencies, make use of x86 instruction set extensions like Streaming SIMD Extensions, short SSE, and its successors, as well as common crypto extensions. Modern ARM CPUs widely used in mobile devices have their own 64-128 bit SIMD instruction set called NEON. It is available since ARM Architecture Version 7 (ARMv7) [163]. The x86 and NEON intrinsic functions are different and there is no one-to-one correspondence between them [163]. Nevertheless, projects like sse2neon [102] offer translations from SSE to ARM NEON intrinsics. As we were not able to port parts of the library and protocol source files to use NEON, the intrinsics are disabled for the Android library port.

5.4.5 *Evaluation Setup*

The metrics used to evaluate the app are runtime, communication, CPU usage, and energy consumption on a smartphone. The evaluation has to cover different scenarios for parameters like network environment and circuit functionality. The impact of different protocol phases is also of interest. The server's set was set to 2^{19} and the client's set to 2^{10}.

For the experiments, a Lenovo Thinkpad T480s laptop with an Intel Core i5-8250U (4 Cores at 1.60-3.40 GHz) and 16 GB of RAM is used as server. The client is a OnePlus 5 phone with Android 9, a Qualcomm Snapdragon 835 Octa-core processor, 6 GB of RAM, and a 3,300 mAh battery. The app is compiled with target SDK version 28 as arm64-v8a application binary interface to match the requirements of the evaluation hardware.

Three different network environments are evaluated using the network emulator NetEm [154] to add additional delay, packet loss, and rate limiting on the server's network interface. In the baseline LAN environment, server and mobile phone are connected to the same local network via Ethernet and 5 GHz WiFi. It has no packet loss, an RTT of 2.49 ± 0.19 ms, 460.1 ± 46.8 MBit/s downstream, and 488.87 ± 70.3 MBit/s upstream. The WAN environment setup represents the case that the mobile phone communicates with a remote server via a stable high-bandwidth connection. Here, packet loss is set to 0.01%, RTT is 40.14 ± 0.7 ms, downsteam is 17.5 ± 3.8 MBit/s and upstream is 17.9 ± 4.1 MBit/s. The RTT and throughput values are aligned to the test setup of Kolesnikov et al. [188]. The LTE environment simulates

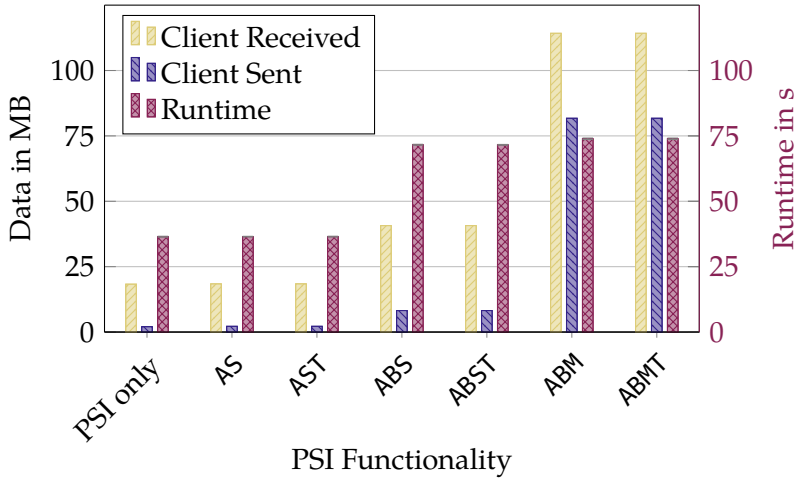


Figure 5.4: Time and data consumption for different risk-scoring functions for calculating the risk score for one day. Runtime and communication data for all different circuits with $n_1 = 2^{10}$ and $n_2 = 2^{19}$ and 2 bit payloads. Runtime means measured with 20 runs each in the LAN setting. The error bars for runtime measurements show the standard deviation. The y-axis for runtime is shown in purple on the right. Figure as in [7].

the setting of a mobile phone connected to a mobile network, which communicates to a distant server over a heavily asymmetric connection. Packet loss is also set to 0.01%, RTT is 50.61 ± 1.65 ms, the connection from server to client has 13.6 ± 2.8 MBit/s while the opposite direction has 3.8 ± 2.0 MBit/s. RTT and throughput values and their standard deviation are measured with the ABY benchmarking tool and iperf3 [167] over at least 20 test runs per data point.

Results on Communication Requirements

As Figure 5.4 shows, complex risk-scoring functionality heavily impacts both runtime and communicated data. The summation of the payload provided by the client does not differ from a circuit that does not perform any functionality on top of PSI (see Figure 5.4, PSI only). When the server also provides payload, runtime sharply increases and the amount of data sent doubles. For the most complex variant of risk scoring following DP-3T or GAEN, multiplications must be added. We can see that while runtime only rises slightly from `ABM`/`ABMT` to `ABS`/`ABST`, the amount of data to be sent increases drastically. For the functionalities `ABM`/`ABMT`, the client must also send and receive more data. As we see in Figure 5.5, the runtime for this circuit is heavily impacted by the asymmetric LTE connection. Revealing results only when a particular threshold value is surpassed has negligible influence on communication and runtime for any circuits.

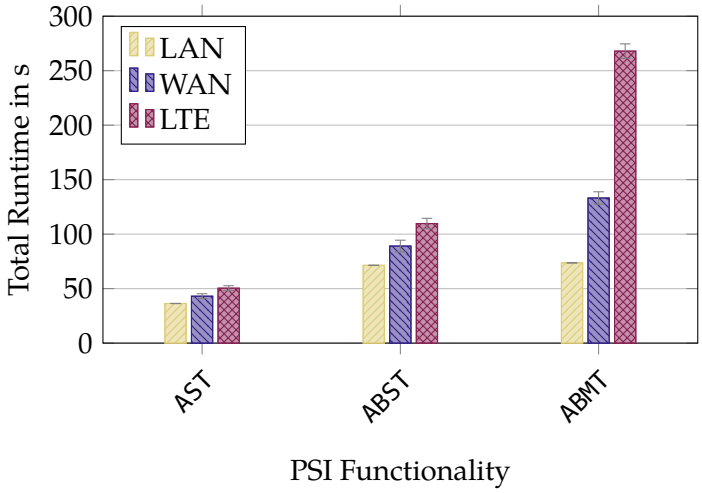


Figure 5.5: Duration of different functionalities for risk scoring in different networks. Error bars show the standard deviation. Figure as in [7].

Standard deviations for the runtime are small within the LAN network. This stability in measurements is caused by a small RTT, high throughput, and almost non-existent packet loss rate in the LAN network. This changes for the WAN and LTE networks, as visualized by the larger error bars. Additional experiments have shown that not emulating additional packet loss does result in a small standard deviation, even for the LTE environment. In general, the runtime increases from LAN to WAN to LTE. The small differences of less than 8 s between the networks do not change for the first two circuits. This shows that including payload from only the client has no impact on runtime across all networks. Once the payload from the server is included, the differences between the networks get slightly bigger (10–20 s). We also conducted experiments with different server set sizes $n_2 = 2^x$ for $x \in \{10, \dots, 21\}$ and $n_1 = 2^{10}$ in the LAN. Both, duration as well as the amount of transmitted data, grow exponentially.

As mentioned, polynomials heavily influence performance. In evaluations of Pinkas et al. for the case of balanced sets, the transport of polynomials required less than 3% of the total communication data [252]. This changes drastically in the unbalanced set case. In our measurements, polynomials are responsible for more than 60% of protocol communication for simple circuits.

Results on Energy Consumption and CPU Usage

Energy consumption is measured using the Battery Historian tool from Google [135], which has been available since Android 5 (Lollipop). With this tool CPU time and estimated battery consumption can be

tracked for an app. The means and standard deviations are displayed in Table 5.2.

Table 5.2: Means and standard deviation for estimated power usage and CPU time measured over 20 runs with $n_1 = 2^{10}$ and $n_2 = 2^{19}$. Table as in [7].

Circuit	PSI only	AST	ABT	ABMT
Est. Power Usage (%)	0.031 ± 0.002	0.034 ± 0.005	0.071 ± 0.003	0.215 ± 0.005
CPU Time (ms)	3507 ± 54	3442 ± 93	4753 ± 74	7960 ± 93

The estimates for power usage are given as a percentage of the battery charge capacity (3,300 mAh) that the app consumed during the execution. The `ABMT` circuit reaches 0.21% of power usage, while the others are between 0.03% and 0.07%. The increased circuit complexity causes a three-fold increase in energy consumption.

The Battery Historian tool also provides an estimate for the power use due to CPU usage. This is reported as 0.00% for all runs, even though Table 5.2 shows that an app execution takes between 3 s to 8 s of CPU time. This value is the sum of the CPU user time and system time. The CPU system time always amounts to less than 30% of the user time. It is unclear whether WiFi energy consumption is accurately accounted for in the estimated power usage for the app. System-level WiFi is responsible for less than 0.03% of energy consumption during app executions.

Results for Different Payload Lengths

To evaluate the impact of different payload lengths, experiments were conducted with $\delta \in \{2, 3, 4\}$ bit payload for an ABMT circuit. Changing from 2 bit to 3 bit, requires the client to send an additional 31.5 KB of data and receive an additional 31.2 KB. Comparing the payload of size 4 bit with the baseline gives an additional 65.0 KB sent and 64.5 KB received. The increase in runtime is minor with around 60 ms from 2 bit to 4 bit payload.

5.4.6 *Privacy and Security Considerations*

This section discusses the privacy of OPPRF-PSI and CERTAIN against semi-honest and malicious adversaries.

OPPRF-PSI

The OPPRF-PSI protocol protects against a semi-honest attacker. This means that no private information is leaked if the attacker follows the protocol. A malicious attacker, on the other hand, might try to deviate from the protocol to either learn private information or break

the functionality of the protocol. To defend against a malicious attacker, all parts of the risk-scoring functionality have to be secured. Pinkas et al. [252] do not propose a maliciously-secure design for OPPRF-PSI. However, they note that modern circuit-PSI protocols based on cuckoo hashing have to rely on the correct hashing of the parties. It is inherently hard to extend protocols based on cuckoo hashing to obtain security against malicious adversaries. This is because the placement of items depends on the exact composition of the input set. Therefore, a malicious party might learn the placement used by the other party[263]. Since OPPRF-PSI applies cuckoo hashing on the client side, this risk exists in case of a malicious server. In [250], PaXoS is used, a data structure for malicious-secure Cuckoo hashing to avoid information leakage. This data structure is not applicable to OPPRF-PSI. The simple hashing into bins could be made more secure with an Encode-Commit scheme as proposed in [263].

Several techniques exist to secure MPC protocols for the circuit phase against a malicious adversary. Among those techniques are cut-and-choose, committed OT, authenticated secret sharing, zero-knowledge proofs, and authenticated garbling [116]. All of these measures heavily impact performance.

The only circuit-based PSI protocol that can be easily secured against malicious adversaries is the SCS protocol[158] by using an additional circuit of size $O(n)$ [252].

As we see, a fully malicious-secure OPPRF-PSI is hard to construct due to the use of cuckoo hashing and the absence of malicious-secure sub-protocols. Switching out some sub-protocols with their malicious-secure variants induces heavy performance penalties. Neither a semi-honest nor a malicious-secure OPPRF-PSI protocol is secure against crafted input sets of either party. Such simple attacks can be made infeasible using mitigation tactics such as rate limiting, threshold circuit functionalities, or device attestation.

CERTAIN

Let us now take a look at CERTAIN as a whole. As described in Section 3.5.3, measures must be taken to ensure that only users with verified diagnoses can upload data. Also, meta-data leakage from communication with the server and attacks on the BLE layer, both discussed in Section 3.6, are relevant for CERTAIN. Additionally, the following threats need to be considered.

An adversarial user might be interested to determine which of the collected pseudonyms belong to diagnosed people. In the semi-honest setting, no pseudonyms of the diagnosed people and no information about the time of encounter are leaked by CERTAIN. This is because only aggregated risk scores are returned to the client. Additionally, inputs of the client and server are protected from the other side by

MPC. Combined, this mitigates deanonymization attacks based on the time of encounter. To gain access to the server's pseudonyms, the adversarial user has to act as a malicious adversary during PSI (see above). To ensure that users do not behave like a malicious adversary, an app attestation mechanism can be used to prove the integrity of the application. To stop adversarial users from repeatedly querying the server with different subsets of their data, a threshold function, which only releases the actual risk value if it exceeds a certain level, is also applied. This measure can be combined with limiting a user's number of queries per day.

An adversarial health authority that is semi-honest does not learn if a querying user is at risk or whom they interacted with because an MPC protocol is used. TEEs and remote attestation mechanisms can be used to ensure that the health authority does not behave maliciously. This allows audits to ensure that the server runs the correct software.

5.4.7 *Practical Discussion*

While CERTAIN provides strong protection for the privacy of diagnosed individuals in a semi-honest setting, the evaluation shows that additional thought has to be placed into the feasibility of the approach.

Risk Scoring Payloads

To produce risk scoring following GAEN v2, the client can pre-compute a risk value according to the duration-at-attenuation for each pseudonym following Equation 5.1 and input this value using only a few payload bits. As we have seen in the experiments, increasing payload size has only little impact on the runtime. For each additional bit of payload length, more data has to be communicated, resulting in a constant overhead of about 62.7-64.75 KB per bit up for relevant sizes, assuming linear growth. In the case of 16-bit payloads, another 906.5 KB of data would have to be communicated between client and server. For an even more fine-grained risk-scoring approach, pseudonym-specific attenuation values could be used as payload.

Real-World Set Sizes

In the evaluation, a server set of size 2^{19} and a client set of 2^{10} were used. Whether or not this is sufficient depends on assumptions on the number of diagnosed people per day and the number of encounters. This value heavily depends on the pandemic situation. DP-3T [305] assumes the number of diagnosed users who upload their data per day to be 2,000. The authors of Epione [304] use 5,000 daily cases for their evaluations. During the height of the pandemic in December 2020, 34,000 new daily cases were registered in Germany [267]. It can also

be assumed that only a fraction of diagnosed people will have the app installed and will, in case of an infection, provide their pseudonyms to the health authority. The German Corona-Warn-App, which builds on GAEN, has been downloaded 28.3 million times as of June 2021 [80]. This is a dissemination of about 34.1% based on the country's total population. Applying this fraction to the number of diagnosed users at the height of the pandemic, about 11,594 users would upload data daily.

Another factor impacting set sizes is the duration of pseudonyms and for how many days in the past encounters with infected people have to be checked. Various approaches use different durations (see Section 3.5.2). We assume that the infectious period is 14 days and pseudonyms change every 15 mins. To provide the user with per-day risk scores, the server set is split into 14 separate sets so risk scores can be computed for each of the corresponding days separately. The day furthest to the past holds the most data (assuming that every day, the same number of infected users are diagnosed and upload their data). For a server set size of 2^{19}, the number of newly infected people that CERTAIN can process is limited to 390 daily new cases. As we can see, there is a mismatch between the real-world numbers and those that can be handled by CERTAIN.

In our evaluation, we used $n_1 = 2^{10}$ for the client. When using the assumptions of the authors of DP-3T that each person collects approximately 140.000 pseudonyms in 14 days, the client's set would be $n_1 = 2^{14}$ [305]. According to Keeling et al. [180], the number of social contacts over a period of 14 days is relatively small. Surveying people in the UK, they deduce that the average number of contacts (independent of duration) is 217 while few individuals will have more than 1000 contacts. Of these 217 contacts, on average, 27% are longer than 15 minutes. This would give us a client-side set of between 2^{10} and 2^{12}. However, the influence of n_1 on the runtime is limited. The number of bins for the client and server is influenced by the bin size on the server side.

Efficiency

To get the total communication data and execution times required for 14-day risk scoring, the OPPRF-PSI evaluation results from Figure 5.4 must be multiplied by 14. Parallelization can be leveraged to decrease runtime. Communication for executing OPPRF-PSI 14 times a day can be up to multiple gigabytes and is, therefore, too high for a system that has to be efficiently scalable. One method to improve efficiency could be reducing the infectious period to 10 days, as proposed by DP-3T [305]. Communication would be decreased by almost 30%, as only 10 OPPRF-PSI instances would be executed each day. A scalability discussion for DP3-T references a 5-day infectious period [305]. This would reduce communication data by around 65% for the OPPRF-PSI app.

By handling input-independent communication, e.g., from the circuit setup phase, differently and reducing the infectious period, the communication values are less impractical but still more than 1 GB of data. As the more complex circuits are a primary contributor to communication, OPPRF-PSI can be used efficiently for DCT, at least in a scenario with 5,000 uploads per day, if advanced risk-scoring functionality is not applied. Another optimization option would be to outsource the client's circuit computation to a set of untrusted independent servers as described by Duong et al. [107]. Performance would significantly improve as these outsourcing servers and the health authority's server would be in a LAN or WAN setting. Also, the amount of data communicated by the client would decrease to $\mathcal{O}(n_1)$ independent of the computed risk-scoring function.

Efficiency is still an issue for CERTAIN that stands in the way of practical feasibility. However, we have shown that providing strong privacy in DCT is possible without leaking sensitive information to the server or the client while maintaining risk-scoring functionalities.

5.5 CHAPTER SUMMARY

In this chapter, two contributions for DCT were presented which leveraged the strong guarantees of cryptographic protocols. In both settings, the health authority or a third party operates the server, which holds data from diagnosed individuals for risk scoring. However, no sensitive location or contact information regarding users trying to determine their risk is revealed to this central entity. Cryptographic constructions additionally ensure that users will not learn more than what is necessary.

The first proposal was to store location traces or BLE pseudonyms from diagnosed users in an ORAM. Users can access the ORAM to query for certain pseudonyms or locations. The evaluation showed that using an ORAM in combination with BLE pseudonyms is feasible. The resulting runtime amounted to less than 1 s for 5,000 new cases per day.

This ORAM-based design and many other DCT approaches, such as the approach of Google and Apple, endanger the privacy of diagnosed persons through leaking timing information. To solve this issue, we presented our second protocol called CERTAIN. It defends against this attack by using circuit-based PSI. Here, the privacy of diagnosed individuals is protected while at the same time providing daily risk scores to users. This proof-of-concept shows that, although the protocol runtime is too long for real-world applications, DCT with risk scoring is possible with minimal leakage to the server and to querying clients. New and faster circuit-based PSI protocols can be plugged into this design to improve performance. An example is the PSI protocol by Rindal et al. [264], which builds on the vector-based construction for oblivious linear-function evaluation by Schoppmann et al. [9] [18].

[18] *The author contributed to the evaluation part of* [9] *as part of her master thesis and is, therefore, listed as co-author.*

This chapter and the previous chapter focused on detecting infection risk using proximity information, either through a BLE pseudonym exchange or via GPS locations. Presence tracing is another tool for pandemic control. It does not focus on the proximity to infected individuals, as in cases of inadequate ventilation in indoor settings, the virus can be transported further than what is considered by proximity detection [81]. Instead, visits to public or quasi-public spaces such as restaurants and concerts are used to determine an infection risk. The following chapter will present several approaches for presence tracing with adaptable privacy guarantees.

6

PRIVACY-PRESERVING SUPER SPREADER DETECTION

This section is based on a paper written in collaboration with Samuel Brack and Björn Scheuermann [3]. It was presented at the ICC COVI-COM Workshop in 2021.

Most proximity-based Digital Contact Tracing (DCT) applications used during the Covid-19 pandemic were built on the Exposure Notification API of Google and Apple, short GAEN (see Section 3.4.3). For privacy reasons, GAEN places the task of individual risk detection to clients. However, due to a lack of centralized data for additional tracing, the distributed approach has sometimes been criticized as not providing valuable data to health authorities [25].

To illustrate this issue, picture a super-spreader event at a restaurant. The super-spreader, an infected person whom we call Alice, uses a GAEN-based contact tracing app. Users of the app who have been near Alice will receive a high-risk warning from the app when the system is informed about Alice's diagnosis. Users seated further away than a specific threshold value (usually 2 m) might receive a weak warning. Due to the indoor situation and, e. g., insufficient air circulation, their risk might be higher than suggested. Users who were out of reach of Alice's Bluetooth Low Energy (BLE) signal will not receive a warning through the app, even if they might be at risk under the given circumstances. Next, assume Alice did not use the GAEN app. In this case, no warnings can be distributed through the app.

As GAEN is insufficient for super-spreader detection, various DCT systems for *presence tracing* were proposed. This idea is related to proximity-based DCT but aims to detect new infections from super-spreader events based on the fact that people have visited the same *location* as a diagnosed person at the same time. For this section, unless noted otherwise, the term "location" refers to a public or quasi-public space like a restaurant, an event space, or a market. Locations are run by an *operator* who answers to the health authority. Common approaches to presence tracing require users to scan a QR code when entering the location. However, each step that requires user interaction dampens usability. By extending a proximity-based system such as GAEN, this extra step can be removed. Providing privacy to undiagnosed users remains the main goal.

In this chapter, we propose two increasingly sophisticated designs for a super-spreader warning system based on presence tracing. The approaches extend the existing GAEN framework with presence tracing. Multiple BLE-capable smartphones, which we call *lighthouses*, send out synced pseudonyms that are recorded by GAEN users. The lighthouses cooperate to cover large (indoor or closely packed) areas. As the distance to lighthouses is irrelevant, only a few devices are necessary to

provide coverage. Setting up this infrastructure is easily feasible due to the large number of old, cheap devices available. The first design relies on a simple broadcasting mechanism to warn users at risk. For the second proposal, lighthouses also collect user pseudonyms and actively check whether standard GAEN has issued warnings for any past visitors. If so, the lighthouses will contact the health authority to upload all relevant recorded user pseudonyms. The design aims to provide usability and privacy, specifically for healthy users. A fallback method is proposed to ensure that app users can be notified even when a potential super-spreader does not use the app.

Our main contributions are:

- Two designs to improve GAEN by handling data regarding visited locations. Users are warned if they have visited a location while a now-diagnosed person was there. No interaction from users, such as a check-in, is required upon entry.
- Only pseudonymous, ephemeral data that does not reveal their history of GPS locations is passed to the health authority by diagnosed users.
- The distribution of warnings does not require human interaction from the health authority. The health authority can manually trigger warnings for diagnosed individuals without the app.
- The functionality of the designs can be tuned according to the privacy need of diagnosed users.

The contents of this chapter are organized as follows. First, related work is presented in Section 6.1. Then, a design based on passive lighthouses is proposed in Section 6.2. The design is extended by actively involving the lighthouses in Section 6.3. In Section 6.4, attacks and corresponding defense mechanisms are discussed for both designs. Section 6.5 presents simulation results. A discussion presented in Section 6.6 considers possible improvements to the system, especially regarding usability.

6.1 RELATED WORK

A common type of presence tracing revolves around check-in systems. Here, users scan a QR code with their presence tracing app when entering (and exiting) a location such as a restaurant or event space. A straightforward implementation of such a system is a centralized database. Real-world apps that followed this schema were Singapore's SafeEntry [286] and the German LUCA app [94]. However, this does not preserve privacy or mitigate data misuse.

Similarly to the broadcast-based apps for proximity-based DCT, New Zealand's NZ Tracer app [222] conducts presence tracing at the users'

end devices. Here, a location operator generates QR codes, which are presented at the entrance. Users can scan the code and store the corresponding information locally. If, during manual contact tracing, the health authority finds that a diagnosed person visited a location, it will publish the corresponding information to all users. The DCT app checks locally if an overlapping stay has been recorded by the user and notifies them if necessary. In case of a warning, the app does not tell users the name of the location. However, a curious, tech-savvy user would be able to identify it.

Another approach for distributed presence tracing is CrowdNotifier [213]. Here, operators of businesses or organizers of events generate three QR codes: one each for entry, exit, and tracing. These codes are created from an asymmetric key pair (pk_l, sk_l) that is derived from a hash of the location's name. Upon arrival, people visiting the location or event scan the entry code with their app. This will locally store a tuple consisting of pk_l, a symmetric notification key, and the current time. Arrival time, notification key nk, and some other parameters are then encrypted by the user with pk_l. The pk_l itself is encrypted with a new asymmetric key pair (pk_u, sk_u) selected by the user. To hide all information, even in the case of forced access to the user's device, the user only stores the following [19]:

$$entry_l = (pk_u, E_{sk_u}(pk_l), E_{sk_l}(nk || \cdots))$$

[19] *As before, $E_k(c)$ represents the encryption of contents c with key k and $||$ is the concatenation function.*

Users can also scan the exit code when leaving, although this step is not necessary. Suppose the health authority discovers during manual tracing that a diagnosed person visited a location or event. In that case, they contact the operator for both the paper lists and the tracing QR code. The tracing QR code contains, among other things, the location name and the notification key, which are encrypted with the public key of the health authority. From this information, it computes sk_l and composes a message m, which is encrypted with the notification key nk. The health authority distributes this (sk_l, m) and the relevant time period to all users. These can then use sk_l to quickly find the relevant entry $entry_l$, deduce the notification key nk, and decrypt the message m. CrowdNotifier leverages cryptographic primitives so only users who have visited the location during the same time as the diagnosed person are notified about the potential outbreak.

Similar results for tracing can be obtained by doing DCT with positioning data. Apps that use GPS data to compare a user's location traces with those of diagnosed individuals to determine who is at risk have been discussed in the previous chapter on cryptographic protocols. Systems that rely on GPS data but have privacy protection through cryptographic techniques are not yet fast or scalable enough for real-world usage. DCT systems that use this data source without cryptographic guarantees generally lack privacy, as they reveal private data of undiagnosed users and their habits to the health authority [78, 232].

A presence tracing app can also utilize Bluetooth or BLE to detect a visit at a location if the location is equipped with suitable senders. In May 2020, Culler et al. [87] presented a presence tracing approach called CoVista that uses BLE beacons to extend the GAEN framework. Their idea is to treat places as people. Our passive approach is very similar to the ideas of Culler et al. They also discuss possible interactions with manual contact tracing. However, our work provides a formal design, parameter analysis, and an extensive security evaluation for the passive lighthouse approach. Unlike Culler et al., we also extend the passive approach by proposing the active lighthouse system, which involves lighthouses in the process of presence tracing.

6.2 PASSIVE LIGHTHOUSES

A successful super-spreader warning system should be helpful to the health authority in containing the outbreaks. It should speed up the health authority's contact tracing and help streamline processes by automating exposure notifications for large amounts of people. User privacy should be one of the leading design goals to ensure that users trust the system and do not avoid or circumvent it. Tools that are perceived as part of a surveillance infrastructure will potentially suffer from low adoption rates and, therefore, limited effectiveness[58]. Following the principle of data minimization, only epidemiologically necessary data should be collected.

The main functionality of the approaches presented in the following is that visitors of locations are notified if their stay has overlapped with the stay of a diagnosed person, even when the proximity-based DCT app did not collect the corresponding ephemeral pseudonyms of the infected person. To receive warnings, users must have the app installed and active during their stay. For usability purposes, no manual user interaction should be required.

The passive lighthouse approach discussed in this section relies on the operator to set up BLE beacons called lighthouses around their location. These send out pseudonyms that are collected by users and uploaded to the existing DCT infrastructure on infection. The essence of this passive approach was first proposed in CoVista [87], but we formalize and extend it in this section.

6.2.1 *Operation*

Operators set up lighthouses, i. e., smartphones with the lighthouse app installed, in their locations. Lighthouses continuously emit ephemeral pseudonyms over BLE, which we call lighthouse pseudonyms or *LP*s. Lighthouses are organized in groups to cover areas larger than the reach of a single device. *LP*s are generated randomly and are distinguishable from BLE pseudonyms broadcast for the proximity-based

DCT by an additional transmitted prefix. The prefix is different for each location and fixed. After a specific time T_{duty} a new *LP* is generated and broadcast.

When a visitor stays at a location, their proximity-based DCT app will broadcast ephemeral pseudonyms, *P*s for short, and collect those of other users. Visitors will additionally collect *LP*s transmitted by the lighthouse. In broadcast-based DCT approaches like GAEN, a user uploads their past *P*s (or the corresponding key material) after being diagnosed. For our super-spreader warning system, the diagnosed user will additionally upload all *LP*s they have recorded during the relevant time period. Prefixes of the *LP*s are removed before upload. Users can opt out of uploading certain *LP*s. The health authority broadcasts both *P*s and *LP*s to all users. These check locally if any of the *LP*s (and *P*s) they have recorded in the past matches. If a match is found, the user is automatically notified that their visit to a location had overlapped with that of a diagnosed individual. More specifically, the user will learn only when they could have gotten infected. Users who did not visit the location or visited at a different time will not receive a warning. Since risk assessment is done locally, the health authority does not learn the location history of users and cannot identify users at risk.

Pseudonym rotation in BLE originates from the need to protect the privacy of device owners. This is not necessary for lighthouses. Therefore, T_{Duty} can be significantly longer than the standard rotation periods used in proximity-based DCT. The decreased number of *LP*s reduces the system's load as users upload fewer *LP*s for the same visit to a location. To mitigate false-positive visits, visitors only store *LP*s if they have received them for a duration T_{thres}, e. g., 10 min. This way, people passing by a location are not warned by accident. Unlike proximity-based DCT, distance information can be ignored for *LP*s during risk assessment. To improve performance, an idea mentioned by the authors of DP-3T [305] can be used. The health authority stores all uploaded pseudonyms in a hash table. After downloading the table, users can check if their recorded *P*s and *LP*s cause a hash collision. To keep the failure probability low, the health authority has to create a new table after some time.

In case the health authority discovers during manual contact tracing that a diagnosed person visited a location, this information can also be fed into the warning system. Using a low-latency, commonly available channel like the telephone, the health authority contacts the location operator and asks them to upload the *LP*s for the corresponding time period to their servers. The location operator needs a single-use token for uploading, which the health authority can provide over the same communication channel. This process prevents misuse through operators and ensures only locations with confirmed diagnosed cases can upload *LP*s.

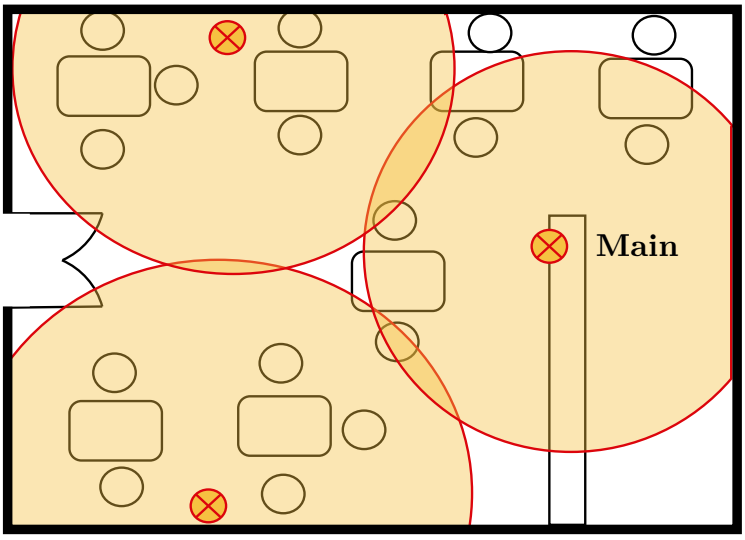


Figure 6.1: Example setup in a restaurant setting with one main lighthouse and two helpers. Note that two visitors in the top right corner are not covered. This can be solved by improving the layout or adding a lighthouse. Figure as in [3].

6.2.2 *Combining Multiple Lighthouses*

If an infrastructure only consists of a single lighthouse, not much is gained compared to proximity-based DCT. Users that see the lighthouse are also likely to see each other. However, especially for indoor locations, the affected area can be larger than the reach of the BLE signal of a single device. For this purpose, multiple lighthouses can form a group to synchronize their *LP*s. One main lighthouse creates *LP*s and communicates them to multiple helpers. If the operator wants to cover multiple floors of their location, they can set up one group of lighthouses per floor. Communication between lighthouses can build on common chat programs like Signal [307] in case an Internet connection is available. An offline solution using Bluetooth pairings can also be implemented as a backup. In such a case, the main lighthouse displays a QR code that the operator scans with the helper devices to establish a connection over Bluetooth or other local channels, like Wifi Direct. An example scenario for a lighthouse infrastructure with helpers is shown in Figure 6.1.

6.2.3 *Subsequent Arrivals and False Positives*

It might be helpful to be able to warn people who arrived shortly after the diagnosed person left. Depending on the virus's durability and the location's ventilation, new arrivals might still be at risk of getting infected [81]. For this reason, a long duration T_{duty} is convenient. If a diagnosed person left during the beginning of the duty cycle of an LP_t,

but stayed long enough for T_{thres} to be surpassed, users that arrived towards the end of T_{duty} of LP_t will receive a warning. In case the diagnosed person leaves towards the end of T_{duty} of LP_t, users who arrive during the cycle of the following pseudonym LP_{t+1} will not receive a warning. To fix this problem, duty cycles should overlap so that for a certain period, two *LP*s are advertised. The overlap $T_{overlap}$ has to be at least as long as T_{thres}.

Another problem with the passive design is that people who have left before the diagnosed person arrived but recorded the same *LP* will also receive a warning even though their risk is minimal. The longer T_{duty}, the more people will receive a false warning if a diagnosed person arrives towards the end of the duty cycle. Therefore, duration T_{duty} should be short. As we see, this optimization criterion contradicts the one discussed above. For this reason, the Simulation in Section 6.5 aims to find a compromise for the optimal length of T_{duty}.

6.3 ACTIVE LIGHTHOUSES

Keeping the false-positive rate as low as possible is essential so users are not flooded with false warnings. To only warn people who were present during the stay of the diagnosed person and thereby minimize the false-positive rate, lighthouses need to become actively involved.

6.3.1 *Operation*

As before, visitors' proximity-based DCT app sends out pseudonyms *P* so other users and lighthouses can record these. Setup and operation are similar to the passive lighthouse system (see Section 6.2.1), with only minor differences. When a lighthouse and a visitor receive the other's pseudonym, they both generate a shared secret *S* by using *P* and *LP* as input for a Diffie-Hellman key exchange [99] [20]. To ensure that the *LP* can not be derived from only *S* and *P*, some additional information known to user and location needs to be incorporated in the secret. Some examples for such a seed are the users BLE MAC address, the location's corse GPS location or the location's static prefix. The lighthouse will store *S*, *P*, and timestamp *T*. The visitor only needs to store *S*.

When a user is diagnosed, they upload all their past pseudonyms *P* to the health authority's servers. They additionally upload all secrets *S* they generated during their contagious period. These will not be made public by the health authority. Master lighthouses regularly check the information broadcasted by the health authority regarding which pseudonyms *P* belong to recently diagnosed people. If the main lighthouse recognizes a P_i from its history in the broadcast, a diagnosed person has visited the location recently. If this happens, the main lighthouse directly contacts the health authority. To prove to the health authority that it can provide meaningful data, the lighthouse will authenticate itself

[20] *The recommended key length for RSA-based Diffie-Hellman used by the active lighthouse system is too large for BLE advertisements. For this reason, Senger [280] suggests using Elliptic Curve Cryptography (ECC). As discussed in Section 4.2.2, `NIST P-224` with some additional twists can be used to transmit an ECC public key in one BLE packet.*

by presenting the corresponding secret S_i. The health authority checks if a diagnosed user uploaded S_i and verifies that no other lighthouse has presented this S_i before. If the provided information is sufficient, the main lighthouse is allowed to upload all pseudonyms P of visitors that had an overlap with the diagnosed person's stay. More specifically, it determines the first and last time when P_i was recorded and uploads all P that fall into this period. A pseudocode representation is shown in Algorithm 1. It can also be helpful to upload some P that have been recorded shortly after. If no information should be leaked about the location of the lighthouse and thereby about the location history of the diagnosed person, all communication with the health authority needs to be conducted through an anonymization service such as Tor [301].

Algorithm 1 Active Lighthouse Algorithm

while true **do**
  $e \leftarrow$ current epoch
  Advertise Lighthouse Pseudonym LP_e over BLE
  $LPs \leftarrow LPs \bigcup \{LP_e\}$
  $P \leftarrow$ Pseudonym received from users
  $S \leftarrow \{\text{Diffie-Hellman}(LP_e, P)\}$
  $T \leftarrow$ Timestamp
  $H \leftarrow H \bigcup \{\text{P,S,T}\}$
  $I \leftarrow$ Data from health authority's broadcast
  if $\exists i := (P_i, S_i, T_i) \in H : P_i \in I$ **then**
    $R \leftarrow R \bigcup \{\forall u \in H : \text{overlap}(i, u) > \text{threshold}\}$
  Send R to health authority, use one S_i for authentication

The active approach ensures that only users who have visited a location simultaneously or – if desired – shortly after a diagnosed user will be informed about their increased infection risk. They will not learn the pseudonym of the diagnosed individual that caused the alarm unless they have come in close contact and have recorded the corresponding pseudonyms P from the diagnosed person. Users who have not visited the location or left before the diagnosed person arrived will not learn that there has been a (potential) outbreak.

It can happen that a later-diagnosed person who does not use any DCT app visits a location, making them undetectable for lighthouses. If the health authority discovers such a case during manual contact tracing, it asks the location operators to upload data for all present users. For this purpose, the corresponding time range and a single-use token are passed on to the location operator. The operator manually inserts both in the main lighthouse, which will use the token to authenticate itself with the health authority and upload all recorded user pseudonyms P from the requested time range.

Similarly to the passive design, a private communication channel has to be established between lighthouses as described in Section 6.2.2.

This channel is used by helper lighthouses to report recorded tuples of (P_i, S_i, T_i) back to the main lighthouse. The main lighthouse stores all recorded data and takes responsibility for communicating with the health authority.

6.4 PRIVACY AND SECURITY CONSIDERATIONS

In this section, several attack vectors against the proposed super-spreader warning systems are discussed to understand and manage potential threats to security and privacy. Attacks on general broadcast-based DCT apps are not considered here. Only new attack vectors introduced by the lighthouse warning systems are analyzed. For mitigation strategies of common threat vectors, such as relay and replay attacks or network observers, the reader is directed to Section 3.6.

6.4.1 *Location Privacy of Undiagnosed Users*

It would be harmful if an adversary could use the lighthouse warning system to learn the location history of arbitrary users. Undiagnosed users never upload any data, so their location history is only leaked if an adversary were to access their device. This is one of the reasons, among others, why *LP*s need to be rotated regularly.

Let us take a look at the active lighthouse system. As long as the adversary, i.e., the health authority, cannot link pseudonyms to people, the location privacy of healthy users is ensured. However, if this assumption does not hold up, various sources of privacy leakage become relevant.

A malicious entity who wants to verify whether an undiagnosed target Tiffany has visited a location can check all *P*s recorded by lighthouses in one location. For this attack, they need to identify which *P*s belong to Tiffany. In GAEN, obtaining key material from undiagnosed users used for deriving *P*s requires either the device to be hacked or physical access by an adversary.

If an active lighthouse detects a past visit of a diagnosed person and uploads all pseudonyms *P* of people at risk in one single message, sensitive information can be leaked. By using additional background knowledge, the health authority can derive that some users visited this location together. To break this link between pseudonyms of undiagnosed visitors, a blind signature scheme similar to the one used in our work CAUDHT [1] can be leveraged, see Section 4.2.

6.4.2 *Location History of Diagnosed Users*

As long as the health authority only learns which locations were visited by diagnosed users, there is *no privacy loss compared to manual contact*

tracing. However, the presented system can be tuned to hide even this data. Let us look at how the location history of diagnosed users can be protected.

First is the passive lighthouse design. Here, only pseudonymous *LP*s, stripped from all static prefixes, are sent to the health authority by the diagnosed user. This stops the health authority from linking locations to *LP*. Additionally, it is crucial that *LP*s are derived locally by lighthouses, are changed frequently, and do not contain hidden information about their creator.

A malicious health authority can misuse the extension that allows sending warnings caused by diagnosed people without a DCT app. By continuously issuing requests to locations to upload their *LP*s, the health authority can learn *LP*s and identify locations visited by diagnosed users, i.e. to verify a guess. Uploading *LP*s requires manual interaction from the location operator. Such an attack could, therefore, be easily detected and would result in a lack of trust and abandonment of the system by location operators.

In the active design, diagnosed users upload their secrets S. As long as the lighthouse, which also knows S, communicates anonymously with the health authority, no information about the location's nature and the diagnosed user's location history is leaked. The health authority does not know the *LP* from which an S was derived. If the health authority wants to map S uploaded by a diagnosed user to locations, it has to use additional data recorded at the location at the time of the visit by eavesdropping on the BLE band. Placing the necessary infrastructure in all possible locations would be rather expensive. But a health authority can single out certain locations of interest and record *LP*s there. This allows the attacker to identify, based on the uploads, whether a diagnosed individual visited a certain place during a specific time.

6.4.3 *Social Graph Leakage for Diagnosed Users*

In the passive design, the health authority can identify that visits of two users to the same location overlap if they are diagnosed and upload the same *LP*. Such an overlap might indicate that they know each other or are in the same social circles. The health authority also records this information about diagnosed individuals during manual contact tracing. However, since it can leak private information, users can decide not to upload *LP*s from specific locations or times. To hide their identity, a diagnosed user can also use Tor for their upload. This works as long as upload tokens, which are usually required to prove to the health authority that the uploader is diagnosed, are not directly linkable to the user. Some token schemes are discussed in Section 3.5 in Chapter 3. In the active design, knowing two secret keys S_A and S_B, the health authority cannot derive if they were recorded at the same time and location. It can only verify a guess for an *LP* it possesses.

6.4.4 *Fake Outbreaks*

There are multiple reasons an attacker can be interested in faking an outbreak. For example, the health authority or a state organization could employ it for crowd control or a competitor of the location operator might want to gain an advantage. In the passive design, the attacker only needs to record *LP*s of locations and have them published by the health authority. An attacker who does not have the capabilities of the health authority can sneak the *LP*s into the uploads of a diagnosed individual. The active design is not vulnerable to this attack as lighthouses provide a sanity check.

Another goal of fake outbreaks or hotspots can be extortion. Diagnosed users might demand money for not visiting a location or uploading the corresponding *LP*s (in the passive design) or *S*s (in the active design). All systems that utilize location data can make operators the target of such an attack.

6.5 SIMULATIONS

This section was added for this thesis and was not part of the original paper as it contains some results that were produced after publishing.

Due to multiple lighthouses working together, both lighthouse systems can span a larger area than simply having people only use their proximity-based DCT app to detect co-location with a diagnosed individual. While the lighthouses themselves also only have limited reach through BLE, these beacons are not used for estimating the distance and are recorded even if the signal is weak but continuous. Unlike GEAN, the lighthouse systems can incorporate information about diagnosed visitors without an app. This section looks at simulation results to determine its effectiveness compared to only using a proximity-based DCT app like GAEN.

Section 6.2.3 mentions that the false-positive rate is influenced by the duration T_{duty} that determines how long an *LP* is advertised. To analyze this issue, a Python script is used to simulate an 8-hour day at a small location such as a restaurant. The maximum capacity of the location is set to 30 people. The behavior of visitors is modeled by utilizing ideas from *queuing theory*. Inter-arrival times of visitors and stay duration are drawn from exponential distributions with means of 10 min and 60 min, respectively. Measurements were repeated 200 times to derive the 95% confidence intervals. The warning precision is given by the fraction of stays that overlapped with the visit of a diagnosed user Alice for at least T_{thres} divided by all users who were warned because of Alice. Figure 6.2 shows the warning precision for an $T_{overlap}$ of 10 min and 15 min. The graph illustrates that a short T_{duty} with a short $T_{overlap}$ (but at least $\geq T_{duty}$) is preferred.

This analysis was extended by Alexander Brunkow in his bachelor thesis [57] [21]. To determine the effectiveness of the passive lighthouse system, Brunkow modeled three types of locations (a restaurant, a night

[21] *The Bachelor thesis of Alexander Brunkow was supervised by the author. He implemented and evaluated the passive lighthouse system.*

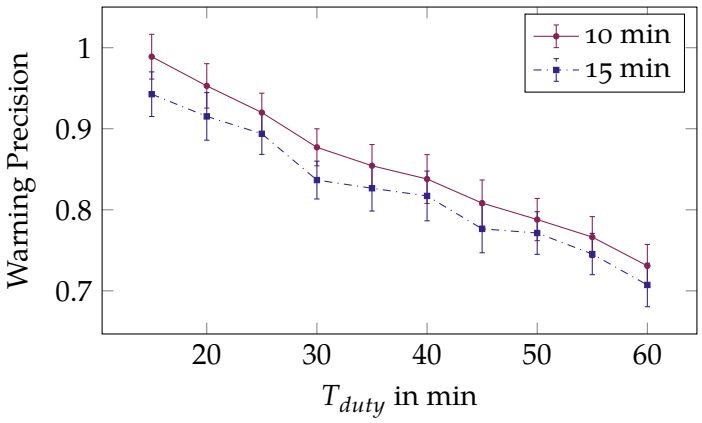


Figure 6.2: Warning precision in the passive lighthouse system for different lengths of T_{duty} and two values for $T_{overlap}$. Figure as in [3].

club, and a grocery store). For each location type, different parameters were used to draw inter-arrival times and stay duration. As expected, locations with longer stay times and lower fluctuation have a lower rate of false negatives. The evaluation for a restaurant setting using different parameters mirrors the above finding that a short T_{duty} is preferable. Brunkow finds that the lighthouse design can be improved if the first *LP* collected at a location is only uploaded by the diagnosed user if they spent a significant amount of time (at least T_{thres}) there. By applying this optimization, the false-positive rate in the restaurant setting decreases from 18 % to ≈0 % in the case of T_{duty}=15 min and $T_{overlap}$=7.5 min and from 22 % to ≈3 % for T_{duty}=30 min and $T_{overlap}$=15 min. The duration of T_{thres}, as well as the question of how long people are exposed to an increased risk even after the infected person has left, depends on the infection risk at a location. This risk is influenced, among other things, by air circulation, mask adherence, and the type of activities common at the location [45]. So this information can be utilized during risk scoring, Brunkow suggests encoding location-specific properties into the *LP*.

Similar to the paper lists required by German restaurants during the pandemic [270], the above simulations only considered simultaneous visits for determining risk. The BLE transmission range of GAEN and the behavior of visitors at the location are ignored. To fill this gap, the master thesis of Alexander Senger [280] evaluates the lighthouse protocol by using *agent-based simulation* [22]. Senger uses the data set of Gabellini et al. [130] to verify the effectiveness of the approach. The data set consists of customers' location traces at four supermarkets of different sizes, covering 26 days each. The data was collected before the pandemic in 2019. It, therefore, does not account for behavioral changes due to lockdowns or contact restrictions. Using these traces, Senger simulates which contacts are detected based on the attenuation of the BLE signal if all customers have the GAEN app installed. Parts of the attenuation are randomized over different simulation runs to account

[22] *The master thesis of Alexander Senger was supervised by the author of this thesis together with Samuel Brack. His thesis focused on the active lighthouse system.*

for a decreasing signal strength due to different holding positions of the device. The detected number of contacts is then compared to the number of people who would not have been warned. As expected, Senger finds that for the supermarket setting, the size of the location and the density of customers play a large role in whether or not an encounter is detected by GAEN. Larger stores with more customers are more likely to result in contacts missed by GAEN. The simulation additionally shows that since the GAEN sample rate is low to safe battery, a certain number of close contacts with a distance of less than 2 m over a period 15 min are missed. The active lighthouse system is able to fix this issue. To ensure that the number of false positives does not skyrocket for large stores, air circulation should be considered when notifying users about potential super-spreader events [45]. Findings from the supermarket setting might not transfer to other cases where customers move around less.

6.6 DISCUSSION

In the following, we discuss the operational aspects of our proposed systems such as recording only relevant lighthouse pseudonyms, integration with other tracing processes at the health authority and deployment costs.

6.6.1 *Collecting Location History of Diagnosed Users*

The lighthouse system can be tuned to match the privacy requirements of diagnosed users. It can either provide full anonymity to diagnosed users or allow the health authority to collect a coarse location history based on uploaded *LP*s. Revealing the diagnosed users' history of visited places to the health authority does not leak more data than what would be collected during manual tracing.

Some locations are more likely to become hotspots for super-spreader events due to the activities practiced there [45]. For example, singing or physical activity are linked to a higher risk than eating in a restaurant. Incorporating such data into tracing can, therefore, increase tracing efficiency.

To facilitate a centralized collection of locations visited by diagnosed users, *LP*s need to contain information regarding the corresponding location that is cryptographically linked to the operator. This can be done through signatures with a key published through a public key infrastructure. To mitigate misuse, only the health authority should be able to read this information. This can be achieved by using an *LP* which is the operator's identity encrypted with the health authority's public key (together with a timestamp).

6.6.2 *Integration With Paper Lists*

Since pseudonyms of diagnosed users are published by the health authority (as by design of broadcast-basted DCT apps), lighthouses can automatically check if a diagnosed person has visited their location. If a visit of a diagnosed person is detected, the main lighthouse can prompt the operator and inform them that they have to provide their paper trail to the health authority. This requires lighthouses to scan the health authority's broadcast for their own *LP*s, as done by the active approach. Having the operator approach the health authority instead of the other way around speeds up detection times. This increases the efficiency of the DCT system as the outbreak might otherwise be detected days later.

6.6.3 *Recording the Correct Lighthouse Pseudonyms*

Assume a location has set up a separate group of lighthouses for each of their floors. A visitor might detect multiple *LP*s at the same time, even those from a group on a different floor. In case a past visitor turns is diagnosed, this could lead to a false warning issued to people who were on a different floor. Therefore, users only record the *LP* that was the closest for at least the duration T_{thres}. If several lighthouses are equally close or the error of the proximity measurement is too large to make a meaningful decision, pseudonyms of multiple lighthouses can be stored. This ensures that movement between locations is also recorded.

6.6.4 *Neighbors of Locations with Lighthouses*

Proximity is only one factor in the detection of a lighthouse, as users will always choose the one that is closest to them. Locations often have neighbors who live next door but might not come in. These neighbors will detect the installed lighthouses and will be warned in case of an outbreak at the location, even though they are not at risk. This can be partially mitigated by setting a threshold for the distance to the lighthouse so that visitors will only consider lighthouses that are less than e. g., 5 m away. To ensure that all visitors can still interact with lighthouses even when seated in a corner, the operator has to ensure good coverage. Another option for mitigating false alarms would be to have lighthouses transmit a static identifier (e. g., a prefix used for forming groups as discussed earlier), which will not be uploaded to the health authority. This allows neighbors to block certain lighthouses for which *LP*s will not be recorded. To make it more easily usable, this could be done with one simple button press, which places all currently received prefixes of *LP*s on an ignore list.

6.6.5 *Usability and Accessability*

The usability is an essential feature of the proposed lighthouse system compared to the check-in approaches. Users do not have to do any scanning when entering a location, record or reveal their GPS traces, but will still receive location-specific warnings. This makes the system accessible, for example, for people who have difficulties using their phones or are visually impaired. For usability reasons, it is also important that the visitor's application can run in the background without draining the device's battery. The passive design without prefixes would not require changes to the GAEN framework. All other proposals discussed in this chapter do require changes.

6.6.6 *Deployment Costs*

Setting up the lighthouse system incurs some costs for the location operator. Apart from the software, each lighthouse requires a smartphone that is recent enough to be equipped with BLE. There is no requirement for specialized hardware, so even second-hand off-the-shelf phones can be compatible with the lighthouse system. To sync LP generation and rollovers, as well as to facilitate the interaction with the health authority, multiple phones in one location need to be interconnected. At least one of the smartphones must be connected to the Internet. Most locations probably have some kind of Internet access, but in some scenarios, this might incur additional costs, e. g., in a long-distance bus where a mobile Internet contract is needed.

Battery consumption is not only relevant for the users but also for the lighthouses. Brunkow [57] implemented the passive lighthouse system with multiple helper lighthouses and one main lighthouse coordinating the *LP* rotation. His evaluations of the battery life of the main lighthouse showed that after 24 h, only 10 % of the battery was consumed. This illustrates that lighthouse deployment is feasible even without continuous charging.

6.7 CHAPTER SUMMARY

In this chapter, we presented a system for sending location-specific super-spreader warnings to users by building on GAEN and similar broadcast-based DCT systems. The proposal extends proximity-based DCT with presence tracing and serves as a tool to deliver notifications of potential super-spreader events quicker than through manual notifications. No GPS data has to be collected as BLE is used to exchange pseudonyms between users and the lighthouses. Multiple lighthouses can cooperate to cover larger areas. The infrastructure of lighthouses is set up by location operators. For this, any off-the-shelf smartphones with BLE capabilities can be used. The lighthouse system warns users

about diagnosed individuals even if they have not recorded this person's pseudonyms.

We presented two designs with different false-positive rates and privacy guarantees. The first one relies on users to record the lighthouses' pseudonyms. In case of an infection, these are distributed using the existing broadcast DCT infrastructure. In the second design, lighthouses actively communicate with the health authority when they recognize that a past visitor was diagnosed. The system then uploads the recorded pseudonyms of everyone whose visit overlapped. Both designs are compatible with GEAN and only require minor changes in the existing code.

By extending DCT with location information and providing an approach that can be adapted based on different privacy requirements, we provide a versatile system for detecting and notifying users of their potential infection risk. The system can be adapted to place different levels of trust in the health authority. On the one hand, it is possible to only inform users of their risk due to their presence at a potential super-spreader event. On the other hand, it is possible to speed up super-spreader detection for newly diagnosed people by actively involving location operators in presence tracing. Here, the privacy of diagnosed users is partially traded for increased tracing performance. This trade-off aligns with requirements for manual tracing, which forces diagnosed users to reveal their recent social contacts and visited locations to the health authority.

INTERLUDE

This part of the thesis examined privacy-preserving methods for combating epidemics and pandemics through contact tracing. To this end, an introduction and overview of proximity-based Digital Contact Tracing (DCT) was given. It was shown that removing trust assumptions regarding central entities, such as health authorities and the government, is crucial for ensuring the voluntary participation of large parts of the population, which is linked to the overall effectiveness of the tracing efforts.

One way to provide provable privacy guarantees in DCT is to conduct risk detection and assessment on users' end devices. This eliminates the need for users to trust a central authority to act honestly. The CAUDHT approach was presented, which leverage this idea by allowing diagnosed users to send anonymous but authenticated messages to their contacts. The second proposal Ovid improves this idea and defends against both misbehaving clients as well as an overly curious messaging server.

It was then studied how cryptographic protocols can be utilized for DCT. Such protocols can ensure that a semi-honest central entity does not learn more information than intended. This allows inherently sensitive data such as GPS locations to be used for proximity-based DCT without leaking users' traces. Additionally, a proof of concept called CERTAIN was presented that mitigates a deanonymization attack that is feasible against all DCT designs that reveal timing information regarding the encounter, which are all approaches using Bluetooth Low Energie (BLE) proximity detection with client-side risk scoring. By eliminating the leakage of the time of the encounter through the use of circuit-based Private Set Intersection (PSI), users who have received a warning can no longer mount the attack.

Next, we examined presence tracing for super-spreader detection and demonstrated how it can enhance proximity-based DCT. By equipping public or quasi-public spaces with BLE lighthouses, the Google Apple Exposure Notifications (GAEN) system can be extended to transmit warnings about potential super-spreader events at such locations. Additionally, tracing efforts can be accelerated by having lighthouses check for warnings regarding past visitors. The proposed systems preserve the privacy of users at risk and can be adapted for different privacy requirements of diagnosed users.

Fighting an epidemic or pandemic involves more than just tracing contacts. In order to make policy decisions, such as determining when a lockdown is necessary, versatile statistics are a crucial factor. Therefore, the second part of this thesis explores how to collect data for

statistical analysis in a privacy-preserving manner from volunteers in distributed settings. To this end, the requirements of existing studies that use mobile devices for data collection are analyzed. By modeling this process, various sources of privacy leaks can be identified. Subsequently, a data analysis platform is presented that minimizes the privacy threat of such data collection campaigns. Its construction mitigates data leaks to malicious data analyzers and to the infrastructure hosting the platform. The platform enables crowdsourcing and data analysis in a privacy-preserving manner. It can be utilized for both pandemic and post-pandemic data collection use cases.

Part III

PRIVACY-PRESERVING DATA ANALYSIS

7

PRIVACY THREAT MODELING FOR MOBILE DATA DONATIONS

This chapter is based on a paper written in collaboration with Björn Scheuermann [8]. It was presented in 2023 at the International Workshop on Privacy Engineering, co-hosted with the Euro S&P.

The widespread use of smartphones and wearables has made it easy for users to continuously collect data about themselves regarding their movement and health [279]. For research purposes, people are willing to share data even if it is sensitive [74]. As a result, an increasing number of studies rely on voluntary *data donations* [276]. Existing platforms like Apple's ResearchKit [36] distribute apps designed by researchers to the public. Here, the responsibility to protect the collected data and the privacy of data donors is placed on the shoulders of researchers. However, healthcare facilities and associated researchers can be hacked or compromised and data breaches have become a frequent occurrence [309]. Legal frameworks such as the General Data Protection Regulation (GDPR) allow the imposing of fines on offenders in case data is lost or misconduct can be proven [132]. Even so, it would be preferable if disclosure of private information could be prevented before it occurs. A large body of research aims to provide technical and statistical privacy guarantees in diverse settings and under various threat models. Tools to this end include Multi-Party Computation (MPC) and homomorphic encryption but also Trusted Execution Environments (TEEs) and local differential privacy. However, these can limit the utility and the expressiveness of the collected data, analysis methods, and final results.

In this chapter, we examine the requirements that a privacy-preserving platform for collecting and analyzing data from mobile devices must fulfill. The contributions of this chapter are:

- A review of 74 existing data donation apps, identifying common functionalities required by medical researchers.
- A model of the parties involved in mobile data donations, taking into account their motivations and goals, as well as their privacy, security, and functional needs.
- Comprehensive data flow diagrams representing an exemplary data donation campaign based on the analyzed apps.
- We analyze threats to privacy, security, and functionality of the existing data donation workflow using the LINDDUN framework. We thereby follow data minimization principles to identify data leaks and privacy threats.

This chapter is organized as follows. Related work is presented in Section 7.1. Section 7.2 analyzes the functionalities of existing data donation apps. In Section 7.3, the relevant parties and their requirements

are modeled. Section 7.4 discusses the LINDDUN Framework as well as the system model we used to represent a common data donation campaign using mobile devices. Privacy and security threats that were identified using this model are discussed in Section 7.5.

7.1 RELATED WORK

Various publications and surveys exist analyzing the usefulness of apps in health care and research. Schmitz et al. [276] analyzed 36 study apps for health research to evaluate the possibilities and challenges provided by mobile health research applications. However, privacy was not their main target. Aljedaani et al. [26] systematically reviewed the security of research-related mobile health apps. Security is a precondition for privacy, which is the main focus of this work. Nurgalieva et al. [242] analyze health apps focusing on security and privacy. While they do propose best practices, they do not take a systematic approach to model the system or discover threats. Iwaya et al. [168] use the LINDDUN framework to analyze mental health apps from the Google Play Store. Unlike this work, they also do not model the underlying system but instead use static and dynamic analysis to detect potential threats, which they then examine with the LINDDUN threat catalog.

A limitation inherent to this and similar works on threat analysis is that it relies on the intuition of the threat analyst, even if a systematic approach is used. This can cause threats to be overlooked.

7.2 RESEARCH USING MOBILE DATA DONATIONS

Using mobile devices in data collection campaigns for research purposes has many advantages over conventional study designs [276]. Potential participants are easier to reach if the geographic location is not a barrier. It is also a simple way to conduct studies that monitor behavior or habits over time. Additionally, shorter data collection intervals are feasible and built-in smartphone sensors allow for objective measurements. Problems with mobile studies arise because non-sensor data is collected by the study subjects themselves, making it subjective and in some cases unreliable.

7.2.1 *Functionalities used in Practice*

To understand which functionalities a privacy-preserving data donation system has to provide, the existing scientific literature on studies using mobile devices for collecting sensitive and medical data is analyzed. To this end, we queried the medical publication platform PubMed [233] for clinical trials focusing on mobile health using smartphones and apps. This yielded 74 apps. We first identified categories of app func-

Table 7.1: The most relevant functionalities used by data donation apps. A total of 74 apps from the digital library PubMed [233] were analyzed. Table as in [8].

Category	Count
Informing and educating	45
Self-tracking	44
Reminders and notifications	37
Feedback to participant	29
App interaction	24
Questionnaires	23
Communication with professionals	19
External sensors	10
Wearables	9
Camera	7
Communication between participants	6
Habits in the digital realm	4
Gyroscope	2
GPS	2
Pedometer	2
Sound	1

tionalities. In the next step, for each app, the provided functionalities were analyzed. Multiple categories per app were possible.

The literature review revealed that the following functionalities are relevant to researchers (see Table 7.1 for a quantitative overview). The most required feature is to inform and educate participants about the study and the studied health issue, as well as provide self-help information.

Also noteworthy is the self-tracking of study participants to collect a history of data on symptoms, triggers, medication, quality of life, and other subjective measurement. Here, the focus lies on collecting a small number of measurement continuously in regular intervals. Self-tracking is directly linked to providing feedback to the study participant, for example, about the progress made. If an app requires study participants to manually enter values measured by external (unconnected) instruments on a regular basis, the app also falls into this category.

Closely related to self-tracking are questionnaires. However, compared to self-tracking, questionnaires allow for more complex questions and a larger number of questions. Here, the focus does not lie on providing feedback to study participants. Rather, questionnaires are an evaluation method for the researchers. Study participants can be asked to complete questionnaires once or multiple times during the study period.

Half of the analyzed apps provide feedback to study participants using the supplied data. The nature of this feedback is diverse. Some apps visualize the collected data, while others use notifications, for example, to inform participants how many calories they have left for the day. Sending push notifications, for example, for reminders, is a feature many apps use.

Aside from manual data collection through questionnaires and self-tracking, one third of the apps require study participants to interact with the app, e.g., for experiments, tasks, training, or games. Some apps in this category provide direct feedback to study participants to help them understand their mistakes and progress.

Another category is apps that use external sensors to collect measurements from study participants. These sensors are paired with the mobile device to directly transfer measurements to the study app. Note that apps that use off-the-shelf wearables were considered separately. Both approaches for collecting objective measurements turned out to be almost equally important. Surprisingly, only a few apps require access to the mobile device's internal sensors, such as the gyroscope, the GPS, or the pedometer. Apps that use one internal sensor often also use other internal sensors. The number of apps that tracked the usage of apps or online behavior of study participants, in short, monitoring their habits in the digital realm, is low in this literature review.

Communication turns out to be an important aspect of data donation apps in the medical field. A quarter of the apps contain features supporting or facilitating communication with professionals such as doctors, nurses, or medical technicians. Apps enabling communication between study participants/people with the same health issues occurred less often. Communication with professionals and between study participants is especially interesting in the context of privacy.

Some studies employ functions described above in combination with conventional or sit-in data collection such as scans, DNA analysis, or ECG. Unlike classical crowdsourcing, study apps in the medical field often intend to provide a simple form of health care for the participants. Some apps also fall under the category of public health intervention that aims to improve the physical or mental health of the general public.

7.2.2 *Methodology*

The literature review described above was performed following the PRISMA guidelines for reporting systematic reviews and meta-analyses [202].

The PubMed search was conducted on February 24th, 2023, and has been limited to publications since 2018. In total, the search returned 339 publications. We analyzed the top 100 publications presented by the platform when sorted by relevance. Of these, two publications were duplicates. Another 20 papers were excluded because no full-text

version was publicly available. An additional four publications were ignored because they either did not present an app or presented apps not targeting patients or their caretakers. The remaining 74 publications included in our review were all peer-reviewed and published. Four publications of these presented two or three apps in the same paper. In these cases, the authors of the respective paper tested the same app with an increasing set of functionalities or in combination with an app for professionals. In our evaluation, we only considered the app for patients or their caretakers in the configuration with the most functionalities.

A reason why only a few of the analyzed apps used internal sensors or tracked online behavior and digital health might be related to the methodology. Only medical apps associated with clinical trials were selected. Clinical trials might not be required for research on mobility and digital health. Research apps that use more internal sensors might also be associated with the fields of psychology and computer science. Such publications may be unlikely to be indexed by PubMed.

7.3 RELEVANT PARTIES AND THEIR REQUIREMENTS

As seen in the prior section, mobile apps can offer a wide range of functionalities for crowd-sourced medical research. The motivations of researchers and donors are a vital part of understanding their security and privacy needs. To this end, a literature review is conducted and contextualized to model the needs of the relevant parties toward a data donation system. Privacy requirements are derived from literature on data-sharing behavior and the assumptions that parties behave in their self-interest. In the following, the security, privacy, and functional requirements of researchers, donors, app store, and professionals as well as their motivations are discussed.

7.3.1 *Researchers*

Researchers want to collect private data from participants for their study through an app on the participant's mobile devices. This *study app* is developed by the researchers or a third party hired by the researchers. It is provided for download on a website or in an app store.

In the first step towards such a study app, the study needs to be designed. While doing so, researchers need the flexibility to select the best study design for their research question. Studies can have various formats, such as questionnaires, continuous measurements of specific data types, or assignments to participants where they have to react to or interact with input. Data collection of studies can occur once or continuously by querying participants repeatedly.

When conducting the study, there are several aspects that researchers must consider to obtain meaningful results [276]. A minimum number of participants is required so that statistics become meaningful. To im-

prove the study's statistical validity, the pool of potential participants should be as large as possible. A wide variety of participants is also necessary. Especially in studies with human subjects, it is often essential to have participants from diverse demographics so that results do not suffer from selection bias. Conclusions drawn from a study involving only participants from, e.g., a specific university might not generalize well. Here, studies using mobile devices provide an advantage to researchers over conventional study designs as potential participants are easier to reach through advertisements on the Internet [294]. Also, app-based data collection can be performed across large geographic areas if no data is collected in a lab or by a doctor.

Rich data is particularly interesting to researchers when trying to understand complex relations. Often, researchers require a large number of data points for specific analyses. Modern methods of data collection, such as self-tracking apps or wearables, also open up new possibilities. However, for data to be useful to researchers, it has to fulfill qualitative requirements. Incomplete, inconclusive, or illogical responses and outliers have to be identified to guarantee meaningful and stable results. Metrics that cannot be objectively measured require special attention during analysis to identify biases. It is also essential that the impact of manipulated or bad data is limited. This means that researchers need to be able to filter data and identify misbehaving or malicious donors to remove their data.

As researchers are required to follow legal guidelines for data protection such as the GDPR [132], it can be assumed that they aim to protect the collected data from unauthorized third-party access. Researchers also have a self-interest in protecting intermediate results and findings until publication. Proper data privacy can also ensure that researchers retain the trust of participants. This is especially relevant if further studies are to be conducted. On the other hand, the researchers' main goal is to conduct a study and focus on the evaluation. They may not be IT experts, so it can be assumed that they do not spend large amounts of resources on privacy or security considerations.

7.3.2 *Data Donors*

Study participants, also called *data donors*, take part in a study conducted by researchers. Through the study app, they provide data they collected themselves. The reasons for data donors to participate in studies are manifold. Apart from financial incentives and simple altruism, data donors might want to improve research on a problem they experience themselves or try to understand the research topic at hand [184, 290]. Data donations can also come from a sense of social duty. Benefiting the public good and a legitimate scientific cause also impacts the decision to donate data [184, 290]. In the case of study apps which also function as public health interventions, taking part in a mobile study can be an

easy and private way to get help [276]. Especially if the target of the study is mental health, downloading an app might be less stigmatizing than going to a doctor. Also, help is immediately available as compared to the long waiting times common in the health sector.

Privacy is an important aspect for data donors. They will not partake in a study if they expect disadvantages or drawbacks due to their participation [184, 288]. Data donors also want to protect their data from misuse, such as unauthorized publishing, selling, or usage for purposes unrelated to the initial study [172, 184]. This can conflict with researchers' interests as they might want to use collected data for further studies, redo evaluations, or share it with colleagues internationally [231]. Data donors expect their data to be protected from unauthorized access after donation. More generally, data donors want to retain autonomy over their data [22, 184, 278]. They might also want to withdraw their consent to data sharing after data has been donated. In countries where the GDPR applies, researchers are required to provide this functionality [132].

The user experience is also an important aspect when donating data [84, 145]. Data donors require simple ways to donate their data and will not spend a long time trying to find relevant studies or figuring out upload processes. The app needs to be easy to use as donors shy away from burdensome processes [281]. Another usability requirement is that the process of donating data with a mobile device should only take few resources and be finished quickly so that the device can be used again for other purposes. While this seems self-evident, it is a crucial part when looking at computation or communication-heavy mechanisms for privacy protection.

7.3.3 *App Store*

It is important not to forget that researchers and data donors need some way to connect. Typically, this is done over a university mailing list or via advertisements. In the context of data donations using mobile devices, the respective app store can fulfill this function. It distributes information about studies and researchers' study apps to potential participants.

The app store is a platform that offers third-party apps to its user base. It is, therefore, not directly responsible for the apps which are available for download. However, to retain the trust of its user base the app store has a self-interest in ensuring the quality and reputability of published apps. For this reason, the app store enforces requirements on new apps that are uploaded. Among other things, this includes privacy policies as well as malware screening. If it distributes (too many) malicious apps, users might switch to other platforms.

We derive some functional requirements the app store itself needs to satisfy in a data donation system. First, it should make it simple for

researchers to announce their study to a broad audience and to address their target demography. It also should inform potential participants about the purpose of the study and the institution collecting the data. Additionally, it needs to provide a form of authenticity to data donors. This means the app store should make it easy for the data donor to identify legitimate studies and institutions. To this end, the app store needs to have the trust of both potential participants and researchers. Researchers might fear their reputation is in danger if they release research apps in an app store with too many bad apps.

7.3.4 *Professionals*

As seen in Section 7.2, some studies rely on professionals as a point of contact for donors. These professionals can be hired through the study or they may be the donors' existing primary care providers. They aim to help donors with their medical, psychological, or technical problems. Since it is their profession, the fact that professionals work for a particular study is not private information. Professionals are unlikely to share personal details with their clients. However, the way they interact with donors as well as how, when, and where they use the study app is sensitive information.

7.4 METHODOLOGY AND MODEL FOR THREAT ANALYSIS

In this section, privacy threats to standard data donation campaigns are described. For this purpose, first the data flows of such an exemplary campaign are modeled. Then, this model is analyzed using the LINDDUN framework to identify privacy threats.

7.4.1 *Threat Model*

For our privacy analysis, we assume that all parties can behave in a malicious fashion. This includes data donors, researchers, professionals, the app store, as well as external third parties such as hackers and network observers. Hackers gaining access to the infrastructure of a party are mostly equivalent to the party behaving maliciously. Using systematic analysis, threats that result from such behavior are identified. We assume that the operating systems of mobile devices and servers are trusted not to upload data to external parties by default. However, all devices and servers are in danger of being hacked. This can happen, for example, through a vendor or a supply chain attack.

Data minimization principles were applied during the analysis to identify data leaks and privacy threats.

7.4.2 *The LINDDUN Framework*

The LINDDUN privacy engineering framework [103] enables an analysis of systems to identify new and unknown privacy threats. The framework separates threats into the following categories: linkability (L), identifiability (I), non-repudiation (N), detectability (D), disclosure of information (D), unawareness (U), and non-compliance (N). To analyze a system with LINDDUN, it must first be modeled as a data flow diagram. Here, entities, data stores, processes, and data flows of the system are identified. Not all aspects of the system have to be modeled. However, all processes where potentially private data is processed, stored, or transferred should be represented.

In the next step, threats are identified. For this purpose, each entity, data store, process, and data flow is checked to see whether one of the LINDDUN threat categories (see above) applies. The categories of unawareness and non-compliance are only relevant for entities. If a threat category can be applied to an element (for example, a data store), the LINDDUN threat tree catalog is used to determine whether it poses an actual threat to the system. Each threat tree represents common attack paths for a specific threat category and element type (either entities, data stores, processes, or data flows). Relevant threats that are identified this way are documented. Assumptions made regarding the system are also recorded. The framework provides a methodology to manage the identified threats. We only analyze an exemplary system, so we do not apply this last step. However, we discuss potential solutions for the identified threats.

7.4.3 *A Model for Common Data Donation Campaigns*

Figures 7.1, 7.2, and 7.3 taken together represent the data flow model of a standard data donation campaign. For simplicity and better comprehensibility, we split the model into three parts. Detailed data flows inside the same zone of trust are only modeled when necessary. The analysis only considers data flows that transfer information between entities during the analysis.

Figure 7.1 shows how the researchers' app is installed via the app store on the mobile device of data donors. Researchers publish their study app via the app store. The app store analyzes new apps to ensure compliance with its policies and to detect if malware was incorporated. While not specifically relevant for privacy, this step ensures security. Data donors search for new apps and download the study app from the app store.

Figure 7.2 shows the basic flow of data between the researchers and the donors after the app is installed. Using the app, donors collect data on their mobile devices. To donate, they upload the collected data to the researchers' servers after authenticating themselves. Feedback for

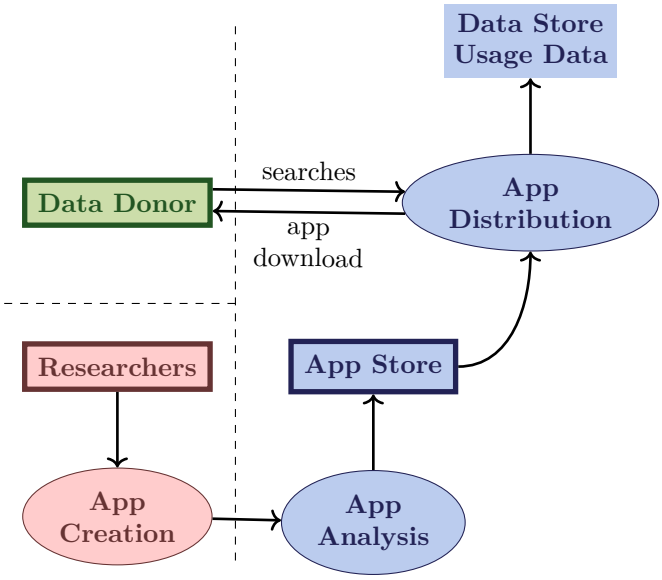


Figure 7.1: Model of the data donation process from app creation to app download by the data donor. Processes under the control of the researcher are highlighted in red. For the donor, green is used, and the app store is blue. Trust boundaries are shown as dashed lines. Figure as in [8].

the donors can be generated based on the uploaded data. Data are processed, stored, and analyzed by the researchers. The results are published or shared with third parties.

As seen in Section 7.2, a significant number of study apps aim to facilitate communication between donors or between donors and (medical) professionals. Both flows are modeled in Figure 7.3. Donors can communicate via the researchers' server with other study participants and professionals. We assumed here that researchers host the communication infrastructure themselves. However, if a third party manages this infrastructure, the same privacy threats arise. The communication is stored on the servers, the donors' devices, and, if applicable, the professionals' devices. To initiate conversations, donors and professionals have to authenticate themselves. Data regarding the communication, such as metadata or contents, can also be donated by the donors. The donated communication data and data collected from the message exchange server can be used during the researchers' analysis.

Assumptions regarding the represented system are made during the creation of the model. First, it is assumed that there is an app signing process. The app store correctly detects malicious apps that contain malware or try to trick donors. Communication between entities is properly encrypted even if messages are exchanged via a platform such as the researchers' server. All private keys are only available to the party that uses them and cannot be derived by another party.

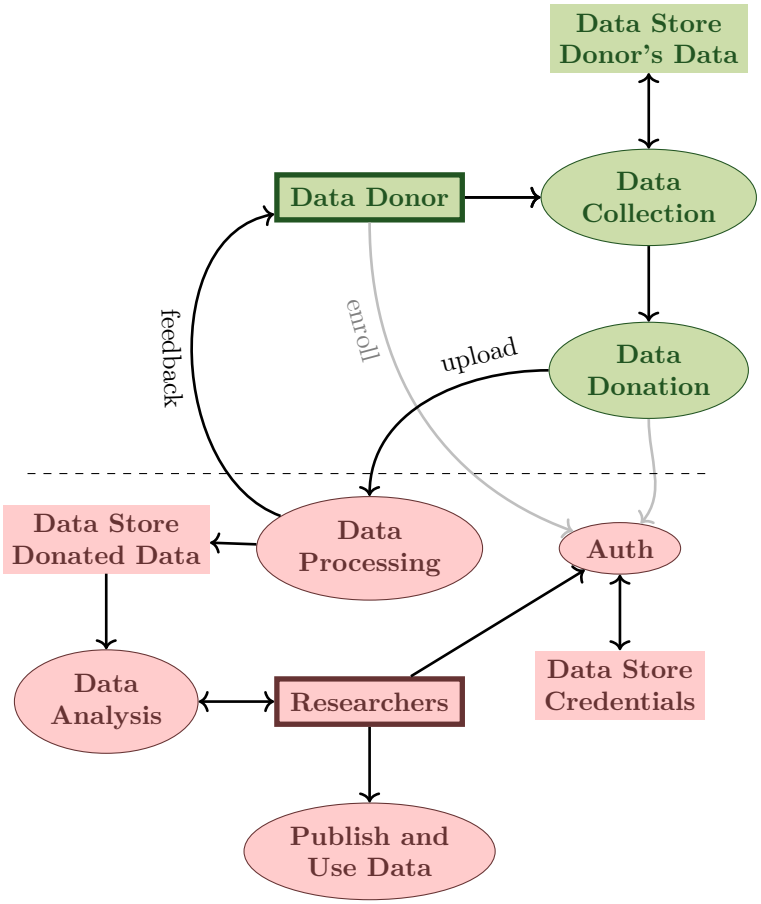


Figure 7.2: The data flow of a minimal study app. Trust boundaries are shown as dashed lines. Figure as in [8].

All data stores are accessible to all internal users. For the researchers' side, this can be a larger number of people. Collaborations between different entities, such as the app store and researchers, are not in the app store's self-interest. However, the app store can assume the role of a researcher.

We do not consider that researchers plan to conduct a particular study as private information. Due to the declaration of Helsinki, it is best practice for medical studies to inform the public about planned studies and their study design before starting [39]. For this reason, it was also assumed that researchers do not make changes to the app while the study is in progress.

7.5 THREATS

In this section, the attack surface exposed by the data flow of standard mobile data donation campaigns is analyzed. This encompasses privacy

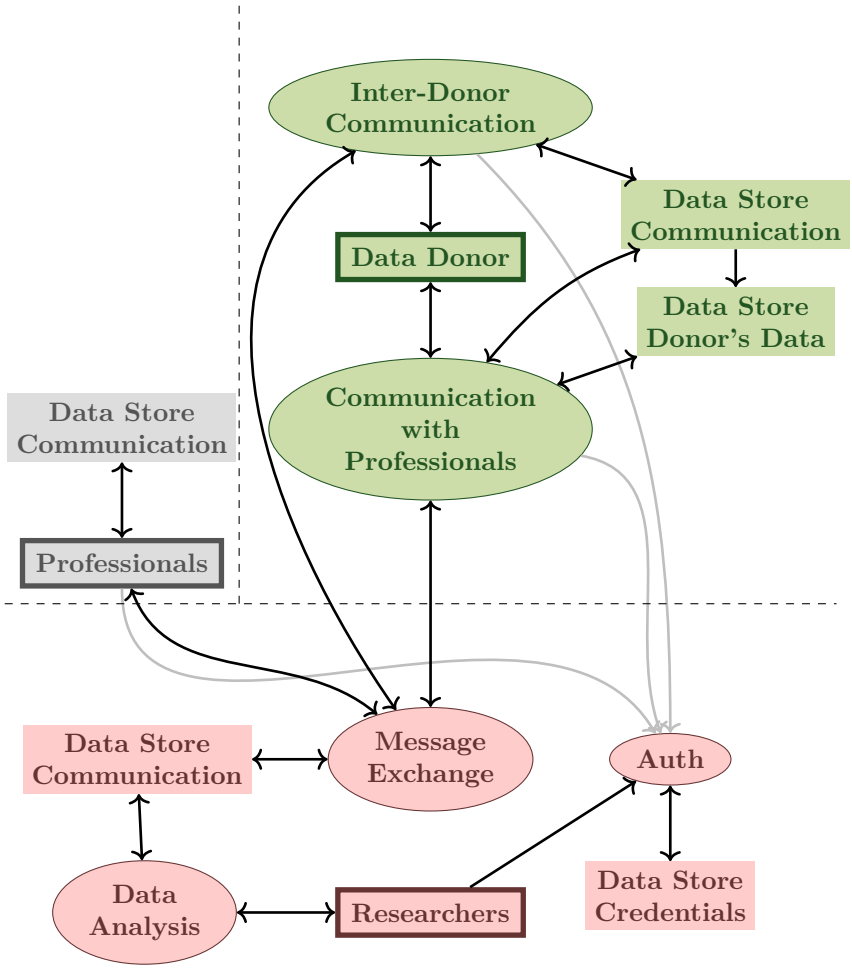


Figure 7.3: Data flow model for a study app that, in addition to the basic functionalities, allows donors to communicate with one another and with professionals. Trust boundaries are shown as dashed lines. Figure as in [8].

threats as well as additional weaknesses that endanger the functionality of mobile data donation campaigns.

7.5.1 *Privacy Threats*

We identified the following 13 threats to privacy using the LINDDUN threat taxonomy. See Table 7.2 for a summary.

Unawareness of Data Sharing

The data collected by the study app from donors, their devices, or their communication might diverge from what the donor expects. This can happen because donors are unaware certain categories of data are collected, stored, and processed at all. They might also not understand how

data is handled. The reasons for this can be diverse. Donors might not read the privacy policies because they are too long[22]. Alternatively, the language used by the privacy policies or the study descriptions might be difficult to understand.

Donors might also provide too much information due to insufficient feedback about which data is collected or due to a lack of user-friendly privacy support. Professionals may also be unaware of the data being collected about them and their communications with donors. Although they themselves are not study subjects, their professional advice, interaction with donors, and messaging behavior are private and can become part of evaluations.

Removal of Data

We assume that researchers follow the GDPR [132] and provide donors and professionals with means to delete their data. However, it becomes more difficult if a person wants to have the data of others deleted because it contains information about them. Once a message has reached the device of another study participant or professional, it cannot be ensured that the relevant data is deleted if this is requested. The data might have already been saved as a screenshot or copied to a location where the study app cannot delete it.

Non-Compliance

Extensive data sharing might also be caused by an insufficient privacy policy that leaves out details. Furthermore, researchers might choose not to comply with the privacy policy and use donated data for other purposes. For example, they might conduct a study different from the one advertised. They could also use the collected data to start their company, similar to the Cambridge Analytica case [77]. Here, data collected via Facebook for psychological research was misused for political campaigns.

Researchers and professionals might try to identify data donors in studies where donors are only identified through pseudonyms. Reidentified private data can be used to harm or violate donors' privacy by being disclosed to third parties. Researchers can also harm the privacy of donors and professionals through inadequate anonymization of published results [44, 106, 183].

Data Extraction

An external adversary can try to extract data from study participants who communicated with others or donated their data. The latter is feasible if the researchers' feedback (see Figure 7.2) is calculated based on data collected from other donors. If no data anonymization method is used, such as Differential Privacy (DP) [108], the adversary may be

able to deduce the responses of other donors by uploading specifically crafted data [23].

If the study app allows interactions between participants, the attack surface increases. An attacker might pose as a study participant to communicate with honest participants and extract information. Both content and metadata can be used to link or identify these participants, which can result in the leakage of medical data. Honest study participants might not be aware that the receiver of their messages cannot be trusted.

We assume that professionals do not reveal private information about themselves in communications with donors. However, the way they interact with donors, as well as how, when, and where they use the study app, is sensitive. This information can be learned by a fake study participant and used against the professionals.

We have not made any assumptions about how professionals are selected for the study. They might be the participant's primary care provider, an outside professional willing to participate in the study, or someone associated with the researchers. An adversary might choose to impersonate a professional to provide bad advice to study participants and exfiltrate data. To combat the threat of fake professionals, the enrollment process for professionals must be well-secured.

[23] *Section 8.1.3 in the following chapter explains DP and how it can be used for removing the contribution of individuals from data sets.*

Leaks from the Data Stores

All data stores in the model are in danger of revealing private information in case of a leak. A leak can occur through a malicious app on the same device, a hack, or a stolen PIN. Access can also be forced, for example, by domestic partners or law enforcement. Gaining access to a donor's data stores can reveal their medical information if they already started collecting data, as well as their communication history with professionals and other donors. Leaks from the data stores of professionals potentially reveal the health problems of all study participants they have been in contact with. We assume that data stores that are controlled by the researchers contain mostly the same data as what is stored in the data stores of donors and professionals, but in larger amounts. Therefore, they need to be properly protected from internal and external unauthorized access.

Verification of Participation

Sadly, password reuse is still a widespread occurrence. An external party can exploit this and try to authenticate at the server for data collection with a username and password from other platforms. The attacker can target a specific person or look for potential targets through a dictionary attack, e.g., by using a database of stolen credentials.

Measures against such attacks include rate-limiting the number of authentication attempts per account and IP. Authentication can also be

Table 7.2: Summary of threats to privacy and functionality. Listed are the parties which are threatened as well as the source. DD - Data Donor, R - Researchers, AS - App Store, P - Professional, H - Hacker, NO - Network Observer. Table as in [8].

Threat	Target	Source
Unawareness of Data Sharing	DD	R,H
Removal of Data	DD	DD,P
Non-Compliance	DD,P	R,P
Data Extraction	DD,P	DD
Leaks from Data Stores	DD,P	R,H
Verification of Participation	DD	H
Leaks through the App Store	DD	AS,H
Leaks through Network Traffic	DD	NO
Donor-to-Professional	DD,P	R,H
Donor-to-Donor	DD	R,H
Message Boards	DD	R,H
Message Types	DD	R,H
Deanonymisation from Logs	DD	R,H
Bad Data	R	DD
DDoS	R	DD,H
Fake Researchers	R, DD	R
Off-Brand Studies	R, DD	AS

realized without password, e.g. through cryptographic security tokens or one-time passwords sent via SMS.

Leaks through the App Store

App stores collect detailed profiles of users of their platform [211]. This includes information on users' search, purchase, and download history. This data is used to, e.g., recommend new apps. The collected data can reveal private information such as medical predispositions and other private information such as location. Depending on the app store's privacy policy, this data can be sold to third parties. This is especially problematic if users are unaware of this data disclosure or if methods and tools for improving privacy are not user-friendly.

Leaks through Network Traffic

It is well known that network providers collect personal data regarding their customers based on their traffic [122]. This is possible because network providers can monitor the traffic in their network. Even if parts of the packets are encrypted, the routing information is transmitted in the clear. By analyzing the traffic flow, a network observer can learn who communicated with the researchers' servers. This metadata reveals who participated in a particular study. This can again be solved by using

cover traffic. However, this might be difficult to achieve in the setting of a study app that is only downloaded by interested parties. Another option to mitigate this data leak is to use anonymization networks such as Tor [301] or more powerful but high-latency mix-networks [98]. Covert channels can also be used to hide metadata from a network observer. Here, the real data is hidden beside or inside other data. For example, using domain fronting [300], the IP of the researchers' server can be the same as the one of another highly popular service unconnected to the study. The traffic of study participants is therefore hidden.

7.5.2 *Privacy Threats due to Insecure Messaging*

In Section 7.2, we have seen that a surprisingly large number of studies employ some form of communication between donors or with professionals. Due to the large number of possible attack vectors when messages are exchanged, this topic is now discussed in more detail.

As show for example by Frost et al. [128], patients can benefit from actively sharing experiences and interacting with people in similar situations through online communities. However, anonymity plays an important role in the decision to share clinical information online. When researchers or a hired third party host a message exchange service for a study, a large amount of communication metadata becomes accessible. Since the 2013 Snowden Leaks, it is well known that communication metadata is private information and can reveal much about the encrypted contents [196]. This means that metadata leakage must also be considered when building an environment where data donors can communicate privately or anonymously. Data donors might be willing to donate parts of their communication data, but not all of it. Especially if a third party handles the message exchange service, e.g., an existing social platform such as Facebook or a popular messaging service like WhatsApp, metadata collected from the study is in danger of being misused.

In case of metadata misuse, the adversary is the researcher, a third party that hosts the message exchange service, or a hacker who gained access to the infrastructure. In the following, we assume that the message exchange service is set up according to the best practices of using end-to-end encryption and authentication between communicating parties. This assumption is made as most papers in Section 7.2 presenting mobile data donation apps did not go into detail regarding their app implementation. A wide array of approaches for private messaging exist. For example, the Signal protocol [307] supports encrypted communication point-to-point or in groups. It provides confidentiality, integrity, authentication, forward secrecy, and future secrecy in case one of the end devices is compromised for some time. Signal provides some degree of message unlinkability as messages are not authenticated with

non-repudiable cryptographic signatures but instead with ephemeral keys and message authentication codes. This allows only the receiving party to verify authorship. However, the facilitating server can still learn who communicated with whom and when.

Donor-to-Professional Messaging

Let us take a look at the case where donors communicate with professionals. Most studies analyzed in Section 7.2 which provide this feature expect donors to only communicate with professionals when there is an acute problem. The only exception was an app that uses the messaging service for professionals to prepare their clients for the next in-person meeting or assign tasks afterward. Both messaging patterns reveal personal information about the donor and, in the latter case, about the professional. An adversary with access to the servers running the message exchange service can easily discover that a study participant communicated with a professional. The fact that a message exists, even if its contents are encrypted, can reveal that a health emergency occurred.

To prevent an attacker from observing these communication patterns,the users' messages can be hidden in *cover traffic*. Cover messages are indistinguishable from real traffic to the observer but are sent randomly. The pattern and frequency of fake messages need to imitate real traffic realistically. A straightforward approach to cover traffic would be broadcasting encrypted messages to all users [307]. Depending on the number of participants, this is a simple but cost-intensive solution. Some protocols, such as Express [113] and Pung [33], provide metadata-resistant communication with formal guarantees. Similar to protocols for private information retrieval [307], they can be computation or communication intensive. This can impact usability as mobile devices are often on metered connections.

Donor-to-Donor Messaging

Communication between donors can be either one-to-one or via a message board. When communicating one-to-one, an adversary monitoring the communication on the message exchange server (such as a hacker, the host, or a malicious researcher) can build a social graph of the donors. This can reveal which people struggle with similar issues and problems. The privacy risks arising from the disclosure of the social graph between the study participants are limited, as it is unlikely that randomly selected study participants already know each other. However, extending a person's social graph with information on the studies they participate in poses a greater risk. This would be the case if existing platforms where study participants already have an account are used to facilitate communications, such as Facebook or commonly-used messaging services. Solutions for making donor-to-donor communication

private and metadata-resistant are the same as those mentioned in the prior section.

Message Boards

Study participants might also communicate with each other via a message board hosted by the researchers or a hired third party. To find new groups and conversations to participate in, the general topics need to be visible to all donors and, thereby, also to the researchers. An adversary with access to the message board's metadata, such as a researcher or a hacker, can observe group membership. This information allows for inferring which topics are relevant to a data donor. If a donor receives an (encrypted) message from a specific group or thread, the topic discussed is likely relevant to them. Broadcast protocols are a solution to hide which topics a donor is interested in. However, this only hides the designated receivers. Also, broadcasts can quickly induce performance issues as donors have to download large amounts of data. A less expensive solution building on the idea of cover traffic would be to have donors join random groups and send cover messages to these groups. Similar to one-to-one communication, private information retrieval methods can be used to hide group membership.

Message Types

Some studies analyzed in Section 7.2 allowed data donors to send images, voice samples, or the configuration of their hearing aids to professionals for examination. These data types differ from standard text messages. The fact that a message with a certain data type was communicated must be concealed as it can leak private information regarding the nature of the conversation. In particular, an adversary on the server should not be able to tell an image from a text message. This can be solved by padding messages if common messages are reasonably small. Another option is splitting data into multiple messages with the same length as performed by Tor protocol [301].

Deanonymisation from Logs

Communications via a message exchange service can be linked by a user ID but also via IP address, session ID, client settings, or behavioral patterns. As network connections and login attempts are commonly logged, messages can be linked even if user IDs are pseudonymous. The history of logins can be hidden through anonymous credentials [98]. These credentials allow a verifier to determine that a person is authorized to use a particular service without revealing their identity. They are also not cryptographically linkable to previous server interactions. IP addresses can be hidden from the researchers by oblivious HTTP where network traffic is routed through an independent third party [298] (similar to

a VPN), with anonymization networks such as Tor [301], or by using more powerful but high-latency mix-networks [98].

7.5.3 *Threats to the System Functionality*

In this section, we discuss additional threats to the data donation model introduced in Section 7.4.3. These threats were discovered during the privacy analysis but do not threaten privacy but primarily security and functionality.

Data donors can manipulate studies by sending flawed or skewed data. Donors or an external party can also conduct Denial-of-Service (DoS) attacks against researchers' servers, for instance, to stop an unpopular study. Vaccination studies have been in the center of misinformation campaigns even before the pandemic [156].

Another potential threat is an adversary who poses as a researcher from a trusted institution when uploading a study app to the app store. This could be done to trick people into participating or to discredit the respective institution or researchers.

The app store can copy study apps submitted by researchers to create off-brand versions. It can also stop researchers and data donors from collaborating by limiting the distribution of a study app or hiding it. This is a DoS attack that is against the app store's self-interest. Therefore, we consider it unlikely.

7.6 CHAPTER SUMMARY

In this chapter, we analyzed the motivations for both researchers and data donors to conduct and participate in mobile data donation campaigns. The literature review showed that privacy considerations play an important role, especially for donors. Building on a meta-analysis of the most common functionalities of data donation apps, we analyzed the privacy of such systems using the LINDDUN framework. Our privacy analysis shows that researchers collect diverse data via an infrastructure that does not thoroughly protect donors' privacy. In particular studies that allow socializing between donors or facilitate communication with professionals need to pay special attention to metadata leakage. When creating a privacy-preserving design, it is important to address these issues and also consider the functional requirements. In the following chapter we will present a technical solution to collect data in a privacy-preserving way.

8

PRIVACY-PRESERVING DATA ANALYSIS

This chapter is based on a paper written in collaboration with Gowri R Chandran, Phillipp Schoppmann, Thomas Schneider, Björn Scheuermann [6]. It will be presented at AsiaCCS in July 2024.

Data donation campaigns, such as those discussed in the previous chapter, are not limited to health data. In 2020, Google published statistics on the impact of the Covid-19 pandemic on user mobility to assist the public in mitigating the virus spread [23]. For meaningful results, the study relied on data collected from Google users. The privacy of these individuals is protected through state-of-the-art *Differential Privacy* (DP) mechanisms. Despite the resulting limitations of DP on accuracy, the mobility study has found widespread use in epidemiological modeling [63, 295], environmental research [197], policy [27], public health [148], and urban planning [151]. While DP protects users' privacy in public statistics, the underlying data is often stored centrally, giving analysts full access to all data. This is convenient but facilitates data breaches as seen in the previous chapter. Analyzers who behave maliciously or who have been hacked can extract and misuse data. Data can also be leaked to third parties such as cloud or network providers. Data breaches and misuse can, in turn, reduce users' willingness to share data [143, 146] [24].

[24] *See Section 7.3.2 for a summary of the privacy needs of people willing to donate private data.*

To ensure the trustworthiness of such data donation campaigns, hardware and software solutions can complement legal measures. *Trusted Execution Environments* (TEEs) such as Intel TDX [164], Intel SGX [83], or AMD SEV-SNP [30], are technical approaches that protect processes or virtual machines from malicious hosts in the cloud and allow clients verify the code and contents of a TEE to establish trust. TEEs rely on the host for paging [25], which exposes access patterns and control flow [239]. *Oblivious Random Access Memory* (ORAM) can hide access patterns and, with some minor changes, can be run in TEEs in a server-only mode, meaning no client-side storage is required. Oblivious data structures like oblivious AVL trees [26] build on ORAM properties and index data so queries can be answered systematically [224, 314]. In the case of database queries, such as point and range queries, the leakage of query-specific volume patterns are an additional threat to privacy as they allow an adversary to reconstruct the database over time [178, 181, 194]. Hiding volume patterns is an open challenge in the setting of oblivious data structures and databases for TEEs.

[25] *Paging is a method for memory management in modern operating systems. Here, data is stored on a secondary storage in memory blocks called pages, which are retrieved when needed.*

[26] *A binary tree construction named after its inventors Adelson-Velski and Landis.*

In this chapter, we present Menhir, a privacy-preserving TEE database. It protects against access pattern leakage and volume pattern leakage in a server-only data collection setting. By leveraging TEE remote attestation for Menhir, data subjects can rely on privacy guarantees against compromised data analysts. They can also be sure that no private data is leaked to the server provider.

Mapping these guarantees to the data donation campaigns of the previous chapter, Menhir ensures data donors can donate their data without researchers or the cloud infrastructure learning private information. While researchers can not learn sensitive data of individuals, they can learn differentially private results. As mentioned above regarding the aforementioned mobility study [23], such results can be very versatile. Data donors can verify the integrity of the Menhir database with TEE remote attestation to convince themselves that their data is protected against extraction attacks and, thereby, against misuse. The information required for remote attestation must be passed to data donors by a trusted third party or must be publicly verifiable to ensure the integrity of the database can be attested.

Menhir requires a server with a TEE such as Intel TDX or AMD SEV-SNP with remote attestation and, depending on the number of database columns, in the order of 2.0 GB of RAM or more for 2^{20} data points. Menhir can also be applied to other TEEs like Intel SGX. Our evaluation shows that insertion operations to the oblivious database are fast and take around 10 ms even on a database with 2^{24} data points.

Menhir allows storing data points with multiple columns in combination with an additional unindexed file with very little impact on query performance. This makes Menhir well suited for crowdsourcing applications. It is also helpful for applications where the stored files must be queried based on certain keys. These files can contain sequential data or additional data fields. For example, in a research study on collecting location traces, the indexed columns can contain personal details, while the file itself is a long list of past locations. By adopting Menhir, privacy can be safeguarded while generating insightful location histograms that consider sensitive information such as infection status or occupation. Furthermore, such files can potentially accommodate more intricate forms of data, including images or voice samples.

The contributions are summarized as follows:

- We present an oblivious database that supports point and range queries, SQL-like `WHERE`-clauses, and differentially private aggregation. Our construction protects against both access pattern and volume pattern leakage.
- We show how data volume pattern leakage can be used to extract data from the state-of-the-art oblivious AVL tree construction Oblix [224] and how protecting volume patterns with differentially private sanitizers thwarts that attack. Our volume sanitizer improves upon prior work [55] by guaranteeing correctness and requiring fewer dummies for the same DP parameters.
- As part of our construction, we provide a multi-index data structure based on AVL trees that is optimized for ORAMs while avoiding expensive constructions such as oblivious priority queues. This is of independent interest.

- We prove the correctness and obliviousness of our construction.
- We published our implementation [27] and provide various benchmarks showing its practicality.

[27] *The source code is available at* `https://github.com/ReichertL/Menhir`.

This chapter is organized as follows. TEEs, the concept of obliviousness, and DP are introduced in Section 8.1. Then, related work is discussed in Section 8.2. The threat model and an overview of the oblivious database construction are presented in Section 8.3. Section 8.4 discusses the details of the construction of the oblivious database and improvements on oblivious AVL trees. This is followed by a performance analysis in Section 8.5. Appendix B contains pseudocode and proofs of the correctness and obliviousness of the Menhir database.

8.1 ON TRUSTED EXECUTION ENVIRONMENTS, OBLIVIOUS ALGORITHMS, AND DIFFERENTIAL PRIVACY

First, TEEs and their problems, such as access pattern leakage, are explained. We describe how ORAM can be generalized to data structures. Oblivious data structures are used by Menhir to defend against access pattern leakage. Menhir also prevents data leakage from volume patterns and through malicious analyzers. To achieve these guarantees, DP plays an essential role in its design. This concept is explained in detail in the last part of this section. For volume pattern sanitation, truncated distributions for noise generation are especially relevant.

8.1.1 *Trusted Execution Environments*

Trusted Execution Environments (TEEs) are a feature of CPUs that provide secure runtime environments. It aims to protect code and data from an adversary on the same system. Some TEEs only shield simple programs, while others isolate complete virtual machines.

Intel SGX is a TEE available in Intel CPUs that aims to protect so-called *enclaves* which run a single program. It suffered from various design flaws and is especially vulnerable to cache-side-channel attacks [239].

Intel Trust Domain Extensions (TDX) is a set of tools supporting virtual machine isolation [272]. It aims to protect the virtual machines called *Trust Domains* (TDs) even if their host is not trusted, i.e., the cloud provider. TDX is designed to guarantee confidentiality and integrity for the memory and CPU state of protected TDs. The TD host cannot access the TD's private memory unless the TD explicitly shares it. A TDX module, supplied and signed by Intel, acts as a trusted middleware between the host and TDs [165]. It provides various middleware functionalities such as interrupt handling. Additionally, it protects TDs from adversaries by recognizing active attacks based on single stepping, page faults, or zero stepping and by keeping branch predictions

from leaking or being tampered with [162]. The *TD owner* is the entity responsible for the software in the TD and updates to it. It, therefore, needs to be trusted. Similar to its predecessor Intel SGX, a TDX TD relies on the untrusted host for scheduling and paging. To run a TD, the host switches to *secure-arbitration mode* and calls the TDX module, which can then create, initialize, and schedule TDs. For paging, the host uses an interface of the TDX module for adding and removing TD pages [162]. Unlike SGX, TDX aims to ensure confidentiality and integrity even against side-channel attacks [165]. To mitigate some types of cache side-channel attacks, a single bit is used for each cache line to signify whether it belongs to a TD. In January 2023, TDX was released on the 4th Gen Xeon Scalable CPU platform. As of July 2024, these CPUs are only available through cloud providers to selected customers [164].

Remote attestation plays a key role for TEEs. This feature allows a challenger to verify that specific trusted software is running inside the TEE. The remote attestation process of TDX first "measures" [165, 272] various integral parts of the system. The resulting measurements allow to verify the integrity of TDX, its components, the corresponding software versions, and loading processes as well as the code and data of the TD. A *quote* is generated from these measurements that is signed and presented to the challenger. The challenger can verify the authenticity of this quote by passing it to an attestation verification service. This service can be run either by Intel or, in the case of remote attestation with Data Center Attestation Primitives (DCAP), by a third party. A relevant part of the quote is a cryptographic hash representing the identity of the TD owner. The TD owner is responsible for the software in the TD and updates to it and, therefore, needs to be trusted.

AMD SEV-SNP [30] is the third generation of AMD's *Secure Encrypted Virtualization* (SEV) TEE that provides trusted virtual machines on AMD server CPUs. It was released in 2020 and leverages existing AMD features for trusted computing like hardware memory encryption, the *Secure Processor* subsystem (AMD SP) for key storage, and encryption of the VM state on context switches. Additionally, SEV-SNP ensures VM memory integrity against a host-level adversary. It also deals with side-channel attacks by restricting interrupts to the trusted VM and protecting branch predictions. Similar to TDX, the host is responsible for scheduling and paging of the TEE VM, which provides a side channel for an adversary with the capabilities of the TEE host.

Neither Intel TDX nor AMD SEV prevent leaks via cache-based side channel attacks from code that performs secret-based memory access, e.g., Prime+Probe [30, 166]. This can be solved by careful programming. Additionally, neither system addresses hardware adversaries or Denial-of-Service (DoS) attacks by the TEE host against the VM.

8.1.2 *Obliviousness*

When processing that that is encrypted or otherwise obfuscated, access patterns to the data can reveal their contents [181, 194]. This is especially relevant, for example, in the case of encrypted cloud databases. To mitigate this leakage, algorithms and data structures can be made (*data-*) *oblivious* so that program control flow and access patterns do not depend on private data [225, 314]. In Section 5.3.1 we introduced the concept of ORAM. In the following, this intuition is generalized.

Adapting the definition of Wang et al. [314], oblivious data structures are defined as follows.

Definition 1. (*Oblivious Data Structure*).
A data structure $\mathcal{D}$ is oblivious if there exists a polynomial time simulator $\mathcal{S}$, such that for any polynomial-length sequence of data structure operations $\overrightarrow{ops} = ((op_1, args_1), ..., (op_M, args_M))$ it holds that

$$addresses_{\mathcal{D}}(\overrightarrow{ops}) \stackrel{c}{\equiv} \mathcal{S}(\mathcal{L}(\overrightarrow{ops})).$$

where $\stackrel{c}{\equiv}$ denotes the computational indistinguishability of two distributions. The physical addresses $addresses_{\mathcal{D}}(\overrightarrow{ops})$ are generated by the oblivious data structure during the sequence of operations $\overrightarrow{ops}$. $\mathcal{L}(\overrightarrow{ops}) = (op_1, \cdots, op_M)$ is the leakage function. It leaks only the operation types and the number of operations, but nothing else.

This definition diverges from Wang et al. [314] to limit the amount of padding required by allowing to leak the type of the operations that are executed.

8.1.3 *Differential Privacy*

Differential Privacy (DP) is an approach for data anonymization that aims to hide the contribution of a single individual by adding well-defined noise to the output of a computation. Let x, y be two data sets where each data point contains sensitive information of a single person. Data sets are called *neighboring* if they differ in at most one item, i.e., the distance between the data sets is $\|x - y\|_1 \leq 1$ for the l_1-norm $\| x \|_1 = \sum_{i=1}^{n} |x_i|$.

Dwork et al. [108] define (ϵ, δ)-DP as follows.

Definition 2. (*Differential Privacy*).
A randomized algorithm M with domain X is (ϵ, δ)-differentially private if for all $S \subseteq \text{range}(M)$ and for all neighboring data sets $x, y \in X$:

$$\Pr[M(x) \in S] \leq \exp(\epsilon) \Pr[M(y) \in S] + \delta.$$

where the probability space is over the coin flips of the mechanism M.

In the case of ϵ-DP (so if δ is zero), it follows from the definition that for every execution of a mechanism M, all neighboring data sets are equally likely to have produced the same outcome. For $\delta > 0$, this property is weakened. Here, (ϵ, δ)-DP allows for the output of a mechanism M to be more likely to be produced by a specific data set x than by a neighboring data set y.

In summary, a mechanism M is differentially private if an adversary cannot tell whether an arbitrary individual was part of the data set. This property can be achieved by adding random noise to the outputs of M. Noise needs to be calculated based on the maximal influence any individual input can have on the output, also called the *sensitivity* Δ of M. The most common approach to generating differentially private noise is to draw noise from a *Laplace distribution* $\text{Lap}(\mu,\lambda)$ [108]. The scale for the noise is proportional to $\lambda = \Delta/\varepsilon$. The value ϵ (as used in Definition 2) is called the privacy parameter or *privacy budget*.

Other variants of DP also exist that leverage different noise distributions. As an example, take a set of values representing the age of a group of students. Here, any negative value, as well as any value larger than 125 years, would be invalid. A *truncated Laplace* distribution allows accounting for this inherent logic of the underlying data [47]. After shifting and discretizing, a truncated Laplace distribution $\text{TSDLap}(t, \lambda)$ has the following properties. Let TSDLap have a support of $\{0, \cdots, 2t\}$ and a probability mass function directly proportional $\exp(-|x - t|/\lambda)$. Bell et al. [47] show that (ε, δ)-DP can be achieved by drawing noise from a truncated Laplace function with $t = \lceil \Delta + \Delta \ln(2/\delta)/\epsilon \rceil$.

An essential property of DP is the immunity to postprocessing [108]. This means that once a differentially private mechanism M is applied, an adversary without additional knowledge of the private data set cannot make the output less differentially private and increase the privacy loss that results from observing a specific output. This property is also relevant when combining multiple (sub-)algorithms that support DP. Two differentially private algorithms can be composed with *sequential composition* if they operate on the same data [108]. The new algorithm then provides $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP. In case two differentially private algorithms operate on disjoint private data sets so that no user is contained in both sets, *parallel composition* can be applied [108]. This results in $(\max(\epsilon_1, \epsilon_2), \max(\delta_1, \delta_2))$-DP.

Depending on the type of data in the data set that is to be anonymized, certain data points might be correlated. This is the case, for example, when two people are walking together or when people from the same family provide data to a genomic data set. To account for such cases, ϵ-DP can be extended to protect the privacy of groups of arbitrary size k by increasing the privacy budget k times [108].

DP has to be applied carefully and is best suited for large data sets. Suppose there are only a few records in a database. In that case, the signal-to-noise ratio can result in a bad utility of outputs as the influence

of a single individual increases, which leads to excess amounts of noise. The contribution of a single individual is usually considered to be a single data point consisting of multiple values. However, defining the sensitivity for anonymization is not as clear-cut in specific data sets, such as location traces or social graphs.

8.2 RELATED WORK

There are many approaches to realizing privacy-preserving data analysis. In this section, we present different approaches to this problem with and without TEEs.

TEEs aim to protect data stored and processed inside as well as the code executed against a malicious host or other types of adversaries on the same system. However, depending on the underlying technology, using a TEE can come with side-channel leakage [239]. At the same time, TEEs provide a great advantage as the performance is better than what is possible with most cryptographic approaches [187]. Also, the guarantee provided by remote attestation can be valuable for systems where trust needs to be well-founded.

A wide range of TEE database systems has been proposed in the past, most of which rely on Intel SGX for data protection. All these approaches provide different database functionalities and security guarantees, especially when it comes to access and volume patterns. ObliDB [114] provides the full range of SQL database functionalities. Encrypted tables are stored outside the SGX enclave in an ORAM and trusted code runs inside the TEE. However, the trusted code does not hide its own access patterns. Also, volume patterns are not considered by this approach. ZeroTrace [274] consists of a secure memory service on top of an ORAM which runs inside an SGX enclave and operates obliviously. However, the memory controller of ZeroTrace does not provide database functionalities. As a result, it does not take volume pattern leakage into account. Oblix [224] uses the methods of Wang et al. [314] to construct an oblivious AVL tree on top of ORAM. The authors pay special attention to the leakage of access patterns from data structures stored inside the TEE. In particular, they discuss how an ORAM client can be changed to not leak access patterns itself. However, they do not consider volume pattern leakage and reveal private information with their query function (see Section 8.4.1). Patel et al. [246] present an approach for volume hiding for encrypted databases without TEE. This approach is similar to the one used by Menhir [55] and protects volume patterns with (ϵ, δ)-DP. They do not consider access pattern leakage.

TEEs are not the only approach to preserve privacy during computation and data analysis. Cryptographic protocols are another solution. Multi-Party Computation (MPC) allows two or more parties to evaluate a joint function over the private inputs of the participants [291] (see Section 5.1). MPC protocols often have quadratic complexity in

the number of parties, so they are not suitable for applications with many clients. In these cases, the computation can be outsourced as proposed by Kamara and Raykova [177] and Prio [82]. MPC protocols do not leak access patterns. They generally compute on all available data, as no branching is allowed. They, therefore, do not reveal volume patterns [123].

Encrypted Search Algorithms (ESAs) are another approach to protecting private data in online databases from malicious cloud services [129]. Here, the data is encrypted, so it is unreadable for anyone who does not have the corresponding decryption key while still retaining the capability to search over it without decryption. This can be achieved, for example, with ORAMs [292], homomorphic encryption [20], property-preserving encryption, or searchable encryption. However, ESAs are prone to data leakages such as access pattern and volume pattern leakage [178, 181, 194].

One main contribution of Menhir is our oblivious multi-index AVL tree with a volume pattern obfuscation. Other approaches exist for obtaining oblivious algorithms or executables. ObliVM [205] is a domain-specific programming language for writing oblivious algorithms. This approach does not provide protection against volume pattern leakage. GhostRider [206] is a compiler that creates an oblivious executable. For this purpose, it employs ORAM techniques. However, GhostRider requires a CPU with a custom co-processor making this approach unsuitable for off-the-shelf TEEs. OBFUSCURO [21] is an obfuscation engine that aims to protect intellectual property by using ORAM techniques to protect program code. While access and timing pattern leakage are considered, volume patterns are not covered by this tool.

8.3 SYSTEM DESIGN OF MENHIR

This section gives a brief overview of Menhir and discusses the security properties of our construction.

8.3.1 *System Overview*

An overview of Menhir's architecture and possible attack vectors is given in Figure 8.1.

Menhir consists of an oblivious database running inside a TEE. As shown in the figure, *data providers* can `insert` or `delete` data in the database. *Data analyzers* can analyze the collected data by issuing queries. To protect the privacy of data providers, the response to each query q is anonymized with (ϵ, δ)-DP using sequential composition. The database relies on an oblivious AVL tree construction for the underlying data structure (see Section 8.4). This construction supports indexing the collected data over multiple columns, which allows queries to filter by these columns. Additionally, an arbitrary file can be stored for each

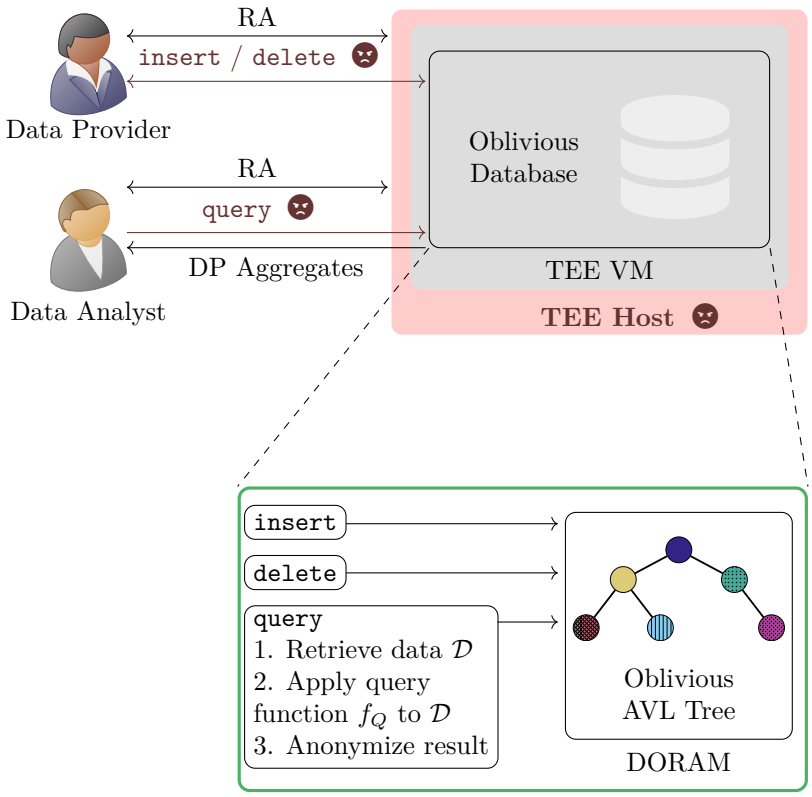


Figure 8.1: Overview of Menhir. Attack vectors are highlighted. RA: Remote Attestation. DP: Differential Privacy. Figure adapted from [6].

data point, facilitating a key-value storage with multiple keys. To protect against a malicious TEE host, all database operations hide access patterns and obfuscate the volume pattern, that is the volume of data required to answer a query.

Each data point consists of multiple keys and a value and is stored in one block of a *doubly-oblivious ORAM* (DORAM) [28]. Each block is part of multiple oblivious trees, one tree per database column. This allows the analyzer to write queries that filter the data by different columns. Accesses to the trees are padded to the worst-case tree height to conceal their structure. When processing queries, all relevant data points are retrieved from the ORAM by accessing the root of the oblivious AVL tree corresponding to the database column being queried. Additionally, dummies are retrieved and processed to hide the amount of data required to answer a query. The number of these dummies is determined through (ϵ, δ)-DP. Before returning the (aggregated) result of a query to the data analyzer, it is anonymized with (ϵ, δ)-DP to protect the privacy of data providers.

[28] *A DORAM is type of ORAM where accesses to the position map do not leak access patterns.*

Menhir extends the previous work Oblix [224] from an oblivious AVL Tree that only protects against access pattern leakage to an oblivious database that supports range and point queries. Menhir additionally protects against volume pattern leakage. For these improvements, sev-

Table 8.1: Comparison of functionality and privacy guarantees between Oblix [224] and Menhir. † Only oblivious retrieval of a fixed number of data points). Table as in [6].

	Oblix [224]	Menhir
Oblivious Data Structure	AVL Tree	Database
Query Functionality	Limited †	More Comprehensive (Point and Range)
Number of Columns That can be Indexed	One	Multiple
Access Pattern Leakage Mitigated	Yes	Yes
Volume Pattern Leakage Mitigated	No	Yes
Volume Sanitation Mechanism	No	Truncated Laplace for (ϵ, δ)-DP
Output Sanitation	No	(ϵ, δ)-DP

eral essential changes had to be made to the underlying oblivious AVL tree construction. One change is to provide a mechanism for retrieving a fixed number of data points from the oblivious AVL tree so that volume sanitation can be achieved. Another change is providing database functionality by constructing multiple AVL trees on the same ORAM nodes. This allows filtering by different columns while minimizing the storage overhead compared to Oblix. Also see Table 8.1 for a comparison of Menhir's functionalities to Oblix.

For sanitizing the volume patterns, Menhir relies on the findings of Epsolute [55]. However, Menhir improves on the theoretical part of Epsolute by introducing the truncated Laplace function for volume sanitation (see Section 8.1.3). This allows dropping the failure probability for volume sanitation which was necessary in Epsolute. Unlike Epsolute, Menhir can ensure that, in all cases, all data points relevant to a query are retrieved and processed.

8.3.2 *Threat Model*

This section describes Menhir's threat model. It considers two different types of attackers with different capabilities: the client and the TEE host.

Client

The client in Menhir can either be a data provider or a data analyzer. We allow the client to be malicious. If the data provider is malicious, they can perform `insert` or `delete` operations to manipulate the database

and the data analysis. By uploading multiple data points, they can skew the evaluation results or mount a DoS attack.

A malicious data analyzer can try to use the `query` interface of Menhir to pose specifically crafted queries in order to reconstruct the database contents. Menhir only allows DP aggregates to be returned by the `query` function to defend against the above attacks. A malicious client, i.e. a data provider or an analyzer, can collude with a malicious TEE Host to infer more information about the data. However, Menhir prevents any leaks resulting from such collusion by hiding access and volume patterns.

TEE Host

In the threat model of Menhir, the main adversary has the capabilities of a TEE Host. The host cannot see the data or code running inside the TEE. However, it can observe the addresses of the accessed data and code at cache line granularity [166]. These access patterns can be used to launch cache-based side channel attacks such as Prime+Probe [30, 166]. In Menhir, we provide protection against these attacks using oblivious data structures, such as the oblivious AVL tree, to hide access patterns.

Our source code carefully implements the presented algorithms by removing data-dependent branching. To ensure no new branching is reintroduced into the final binary through various compiler optimizations [285], a verified compiler such as CompCert [199] can be used, which supports most languages that follow the ISO C 99 standard. Other steps to solving this issue are turning off most compiler optimizations and programming branch-aware code (e.g., by implementing algorithms that are already data oblivious).

Even if the accesses to private data inside the TEE are carefully obfuscated, the number of accesses to a TEE database can reveal the amount of private data that is processed. This information can be used in database reconstruction attacks mounted by the TEE host [181]. For this attack, the TEE host needs to know the column and data interval requested by database queries. In Section 8.4.1, an attack is demonstrated that uses this information on volume patterns. However, for this reconstruction attack, it is also sufficient if the TEE host only has knowledge about how the private data is distributed [194]. We, therefore, assume that the adversary can use the `query` interface of the database and pose maliciously crafted queries. To mitigate the leakage of private information through query results, all database responses are anonymized with DP ("output sanitation"). To defend against database reconstruction attacks using volume patterns, Menhir hides the number of data points processed by a query with DP guarantees ("volume sanitation").

The data points that are inserted into the database are uploaded by potentially untrusted data providers. Therefore, it is possible for the adversary to access the `insert` and `delete` interfaces of the database.

Although we do not consider timing attacks, the mechanism for volume pattern sanitation makes such attacks more difficult. DoS and power analysis attacks as well as attacks requiring physical access to the TEE host are out-of-scope for Menhir.

8.4 OBLIVIOUS DATABASE

In this section, we present the design of our oblivious database system Menhir. The functionality Menhir provides is a database consisting of a single table with multiple rows and columns that allows for `insert`, `delete`, `query`, and pre-filtering similar to SQL `WHERE`-clauses. While support for multiple tables is feasible, realizing privacy-preserving joins poses its own separate challenges [190, 321]. We leave extending Menhir to multiple tables for future work. The supported `query` types are point queries (where all records with a specific key are retrieved) and range queries (where all records falling into a specific interval are returned). Only differentially private aggregates are returned to the analyzer.

8.4.1 *Querying an Oblivious Tree*

Our oblivious database extends the *Doubly-Oblivious Sorted Multimap* (DOSM) construction of Oblix [224]. The DOSM provides the functionality of a key-value store and builds on a DORAM. First this data structure is explained and its data leakage is analyzed. Then, the improved construction called Menhir is presented that fixes this issue.

Doubly-Oblivious Sorted Multimap (DOSM)

Oblivious data structures can be built upon ORAM primitives. Wang et al. [314] show that by replacing pointers in tree-like data structures with pointers to the ORAM, it is possible to make data structures such as AVL trees oblivious. Pointers to child nodes become pointers to the corresponding ORAM block, so $\mathsf{ptr}_i = (ID_i, L_i)$, where ID_i is the ORAM block number and L_i is the corresponding leaf in the ORAM (see Section 8.1.2 in Chapter 2). Oblix [224] transfers this AVL tree construction to a doubly-oblivious ORAM to create a DOSM. The DOSM stores key-value pairs by organizing them as nodes in an AVL tree. When computing operations on this tree, the root node, which is stored separately, is used as the entry point. Depending on the operation, the tree is traversed from top to bottom to either find a node with a certain key or determine the correct location to insert a new node. All such operations must be padded to the worst-case tree height h_{max} so that no information is leaked about the structure of the tree. For an AVL tree with n nodes, $h_{max} = 1.44 \cdot \log_2(n)$. Balanced trees require rebalancing after insertion and deletion operations. These operations also need to be padded to h_{max} and are not allowed to reveal on which level

the insertion or deletion happened. The AVL tree construction allows storing multiple instances of the same key by storing an additional hash to distinguish between data points.

Oblix [224] proposes an algorithm for retrieving several records with the same key. In short, the algorithm DOSM.FindOblix(k, i, j) retrieves all key-value pairs for key k starting from index i to index j. Although not mentioned in the original paper [224], we can see that it is simple to construct full point queries and range queries from this function: for point queries, instead of using the i-th index, always use index zero, and instead of the j-th index use the highest index possible for key k. Similarly, range queries can be constructed by providing different keys for the start and end of the interval.

Leakage through Volume Patterns

In the following, we show how the construction of the Oblix find function DOSM.FindOblix leaks volume pattern, which in turn leaks information about the data points in the database. Following Definition 1 in Chapter 2 on oblivious data structures, the execution of functions on an oblivious data structure is not allowed to leak any private information to an attacker. Let us assume an attacker $\mathcal{A}$ who can monitor the access patterns to code and data. Query responses to this DOSM are computed using the DOSM.FindOblix function. $\mathcal{A}$ does not pose the queries or learn the result. It is sufficient for them to know what ranges are queried so they can observe the resulting patterns. In particular, $\mathcal{A}$ can see the length of the return array of DOSM.FindOblix and use this information to partially or fully reconstruct the DOSM keys. See Figure 8.2 for an example. It shows clearly how insertion operations into the return array provide a side channel to the adversary.

DOSM without Volume Pattern Leakage

As we have seen in Section 8.4.1, the DOSM.FindOblix algorithm of Oblix [224] leaks information via its volume pattern. To mitigate this leakage, the Find algorithm must not reveal the volume of data used to answer specific queries. This has to include temporary data structures like queues and arrays for which the adversary can observe access patterns and volume patterns.

A naïve and information-theoretically secure solution for hiding the volume of the data such that all queries are completely indistinguishable is to process the entire database every time [234]. However, this is extremely inefficient. Consequently, as a trade-off, some information needs to be revealed to the adversary for better efficiency. In the context of distributed ORAMs, Bogatov et al. [55] proposed sanitizing the volume patterns through DP algorithms. Here, using all keys, a hierarchical histogram is created which is then perturbed with DP. We employ this idea and determine the number of dummies d used to

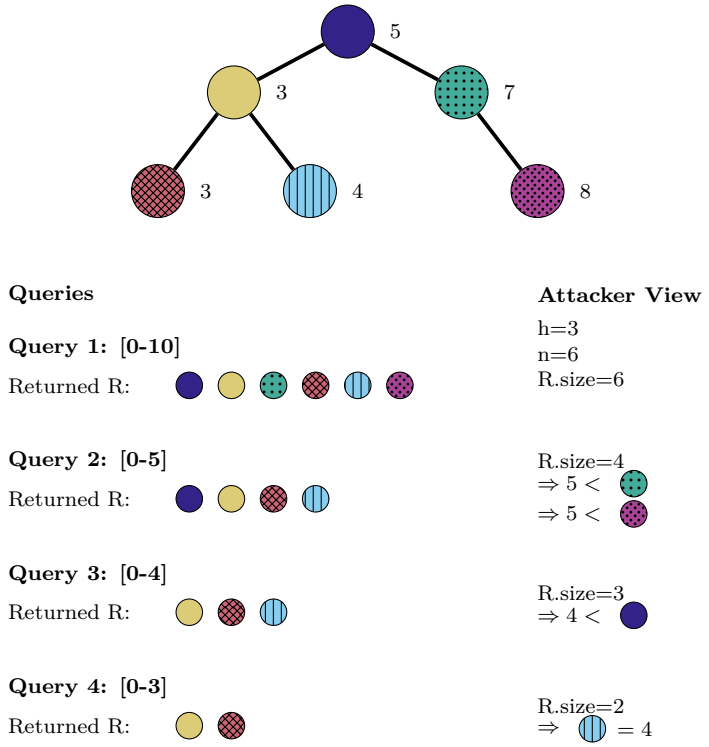


Figure 8.2: Volume pattern leakage of DOSM.FindOblix. Observing the volume of array R allows the attacker to determine the approximate and exact values of nodes. Figure as in [6].

hide the volume of a query with an (ϵ_s, δ_s)-DP sanitation algorithm S. Menhir improves the theoretical results of Bogatov et al. by using a truncated Laplace distribution for sampling noise. This ensures that the number of retrieved data points m is never smaller than the number of nodes n' that need to be returned (we see that: $m = n' + dummies$). The information learned by an attacker who monitors volume patterns is limited through DP. The value of m does not reveal the existence or absence of a node associated with an individual. Optimal parameters for S can be found in [255]. After the data collection has ended and before querying can start, S uses the contents of the DOSM to compute the DP-sanitized volumes for each key. When a query q is posed, the value of m is determined by checking this data structure. For simplicity, we write $m = S(q)$.

To hide the volume pattern, the query function needs to process m values when answering a query. Additionally, the DOSM must be accessed obliviously to not reveal the tree structure. Traversing an AVL tree to sequentially retrieve a fixed number of data points is not straightforward as successors might be located in the right subtree of a node (if it exists) or in a parent node (see Figure 8.3). Unpadded access to the successor can reveal the tree structure and the level at which the node is located. If all accesses are padded, the overhead of additional

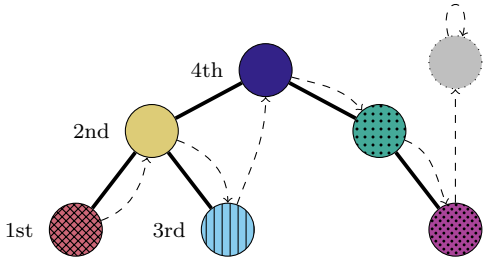


Figure 8.3: Order in which nodes are accessed for the interval $[2, 4]$ and $m = 4$. Pointers to successor nodes are shown with dashed lines. Figure as in [6].

accesses to the ORAM becomes large. As a result, only queries with a small selectivity would be possible.

A better approach is to change the structure of the AVL tree so each tree node holds a pointer to its successor. When a new node is inserted, the pointer of its predecessor is replaced with a pointer to the new node. The new node reuses the predecessor's old successor pointer. We now define the functions for retrieving an interval of key-value pairs and for inserting a new key-value pair into the DOSM. The corresponding pseudocode is given in Algorithm 2 and Algorithm 4 in Appendix B.

- $\mathsf{DOSM.Find}([k_S, k_E], m) \rightarrow [(k_i, v_i)]_1^m$:
  The algorithm is given an interval from start key k_S to end key k_E as well as a fixed number m of entries to return. This algorithm first traverses the tree to find the smallest node for which $k_S \leq k_i$. The number of accesses is padded to h_{max}. Having found this first node, additional $m - 1$ nodes are retrieved and added to the output by sequentially accessing each node's successor. The algorithm returns a set of key-value pairs of cardinality m.

- $\mathsf{DOSM.Insert}(k, v) \rightarrow \perp$:
  The function is given a key k and a value v. First, using k, the tree is traversed starting from the root to find the insertion location of the new node with (k, v). This temporary parent must be a leaf node conforming to the standard AVL insertion strategy. The number of accesses to the ORAM for this step is padded to h_{max}. During traversal, the pointer ptr_{pre} to the predecessor and the pointer ptr_{parent} to the parent are stored. The predecessor is the last node on the way from the root to the leaf, where the path turns to the right child. If the new node is the first node in the tree, no predecessor exists and ptr_{pre} will point to a dummy node (see Figure 8.3). In the next step, the tree is obliviously rebalanced following AVL tree conventions. At last, the successor pointers are updated. If the new node is the first node in the tree, then its ptr_{parent} is used as successor. Otherwise, the new node copies the successor pointer of the predecessor and then sets itself as

successor. If a node does not have a successor because it is the last node of the tree, the pointer will point to a dummy node. The dummy node points to itself.

Using the adapted AVL tree, retrieval using DOSM.Find is possible with $\mathcal{O}(h_{max} + m)$ ORAM operations and insertion using DOSM.Insert can be done in $\mathcal{O}(h_{max})$ ORAM operations.

The **correctness** of the sub-procedure of DOSM.Insert for finding the predecessor of a newly inserted node is given as follows. We call a node smaller than another one if its key is smaller than the key of the other node. In case both keys are equal, a hash associated with each node is used to determine the order. The predecessor of a newly inserted node N_{new} is the largest node which is still smaller than N_{new}. The path from the root to the leaf where N_{new} is inserted consists of a sequence of nodes that are either left or right children. Due to the properties of the binary tree, for all nodes N_r where the path diverges to the right, it holds that $N_r < N_{new}$. This is because all nodes in the right subtree of N_r (where N_{new} is added) are larger than N_r. Any N_r that is found in the right subtree of another N_r is automatically larger and, therefore, better suited as the predecessor for N_{new}. Therefore, the largest node that is still smaller than the new node is the N_r closest to the leaf level. In case the path never diverges to the right and no N_r exists in the path, the new node is the smallest node in the tree and no predecessor exists. Its successor is, therefore, the previously smallest node in the tree. This node is the leaf node that was identified as the insertion location.

As we can see, the number of nodes that need to be altered after a DOSM.Insert is limited and a single rebalancing is sufficient to ensure that the AVL tree invariant is fulfilled. This is due to the fact that all nodes in the tree have balance values of $b \in \{-1, 0, 1\}$ prior to the insertion. The insertion will change this by one. The rebalancing will cause the balance value of the node for which the AVL tree invariant was broken (so $|b| > 1$) to be set to zero. The remaining tree will not become imbalanced if it was balanced before the insertion.

An optimization can be applied during rebalancing. The nodes retrieved during insertion are all nodes from the root to the leaf where the new node is inserted. The rebalancing procedure to be executed depends on the tree's structure. For a left or right rotation, the nodes that need to be updated are the one for which the invariant is broken and one of its children. The imbalance is caused by the newly inserted node. This means both nodes are on the path to the newly inserted node and were retrieved previously. In case a more complex left-right or right-left rotation is required, balancing becomes more difficult. Again the imbalance is caused by the change of subtree heights resulting from the insertion of a new node. Let us take a look at right-left rotations (left-right rotations work analogously). Let N be the node in question, N_r be the right child of, and N_{rl} be the left child of N_r. Rebalancing

procedures that will cause a right-left rotation only occur after insertion to either subtree of N_{rl}. Both N_r and N_{rl} have been retrieved when the new node was inserted. However, as the values of the nodes might have changed during the insertion procedure, they need to be retrieved again. This means for all rebalancing operations at most three nodes need to be retrieved from the ORAM, independent of the type of rebalancing.

8.4.2 *Oblivious Database* (*ODB*)

Menhir can store multiple columns of different types following a column layout schema F. Entries inserted into the database need to follow this schema. It allows for point or range queries along all its indexed columns. Additionally, analyzers can filter along one column. Menhir allows for SQL Queries of the following format

$$\texttt{SELECT}\ f_j(c_f, \epsilon_q)\ \ \texttt{FROM}\ \text{database}\ \texttt{WHERE}\ k_S \leq c_w \leq k_E$$

with query $q \in Q$ is defined as a tuple $q = (k_S, k_E, c_w, c_f, j, \epsilon_q)$. The start key k_S and end key k_E define a range that is used for filtering a column c_w. For the rows that remain after filtering, the values in column c_f are passed to the DP aggregation function f_j. This function is passed in the query via its index j. The function uses privacy budget ϵ_q for anonymization, under the assumption that enough budget is available.

Construction

To create an *Oblivious Database* (ODB), we alter the nodes of the DOSM (see Section 8.4.1) so that each row of the ODB is represented by one node that is stored in the DORAM. For each of the C columns, a DOSM is built using the same nodes. This requires each node to have C pointers to right children and C pointers to left children. The resulting data structure is a multi-index AVL tree. See Table 8.2 for an overview of all information stored in an ODB node. Additionally, a total of C root nodes need to be stored as entry points to each DOSM. We define the ODB as follows:

- $\mathsf{ODB.Init}(N, F) \rightarrow \perp$:
  This function takes as input a maximum number of entries n_{max} and a column schema F with $C = F.size$, i.e., the number of columns in the database. The function then calculates the required block size for the DORAM using the schema F. Next, it initializes a DORAM using n_{max} and the block size. Last, space is allocated for an empty array of length C to store root pointers.

- $\mathsf{ODB.Insert}([k_1, \cdots, k_C], v) \rightarrow (\mathsf{ptr}, h)$:
  This function takes as input a set of C keys $[k_1, \cdots, k_C]$ and one value v. The function computes a hash h. It creates a new node using the provided data and the hash. Then, the tree structure of

Table 8.2: Information stored in the node of an multi-index AVL tree used as basis for an ODB with C columns. Table as in [6].

$\texttt{key}_1, \cdots, \texttt{key}_C$		
value		
hash		
Column 1	⋯	Column C
$\texttt{right\_child\_ptr}_1$	⋯	$\texttt{right\_child\_ptr}_C$
$\texttt{left\_child\_ptr}_1$	⋯	$\texttt{left\_child\_ptr}_C$
$\texttt{left\_height}_1$	⋯	$\texttt{left\_height}_C$
$\texttt{right\_height}_1$	⋯	$\texttt{right\_height}_C$
$\texttt{successor}_1$	⋯	$\texttt{successor}_C$

each of the C DOSMs is updated iteratively. The function returns a pointer to the newly created node as well as a hash.

- ODB.Find$(k_S,k_E,m,c_w) \rightarrow [[k_{i,1}, \cdots, k_{i,C}], v_i]_1^m$:
  On input of an interval $[k_S, k_E]$, the required number of nodes m, and the index c_w of the column to be queried, this function calls DOSM.Find starting with the root node for column c_w. The function returns the keys and values for m nodes starting from the smallest node n_i for which $k_S \leq k_{i,c_w}$. This function is a sub-procedure of ODB.Query and is not exposed to database analyzers.

- ODB.Query$(k_S, k_E, c_w, c_f, j, \epsilon_q) \rightarrow f_j([k_{i,c_f}]_1^m)$
  On input of a query $q = (k_S, k_E, c_w, c_f, j, \epsilon_q)$, determine the value of m for the interval $[k_S, k_E]$ and column c_w using the corresponding volume pattern sanitizer, so $m = S_{c_w}(q)$. Next, ODB.Find is called with the parameters k_S, k_E, m, and c_w. The call returns a set R of rows. Using the data for column c_f of all rows in R [29], function f_j is computed with privacy budget ϵ_q. Then, the aggregated and anonymized query result is returned.

[29] *So k_{c_f} or alternatively v_i, if it contains numeric data.*

- ODB.Delete$(h) \rightarrow \perp$:
  Upon receiving a hash h, the corresponding node n_d is searched. This takes h_{max} accesses. The node N_d is removed from the DO-RAM. Then, all C DOSMs are updated. For each DOSM c, a replacement is found for key k_c of N_d. No replacement is necessary if N_d does not have any children in the DOSM for column C. If N_d has only one child, this child is the replacement. If N_d has two children, the smallest node of the right subtree is used as a replacement. The search for the replacement is padded to h_{max}, independent of the number of children of the deleted node. Once a replacement is found, all nodes in the DOSM on the path to the deleted node are updated. Again, this operation is padded to h_{max}.

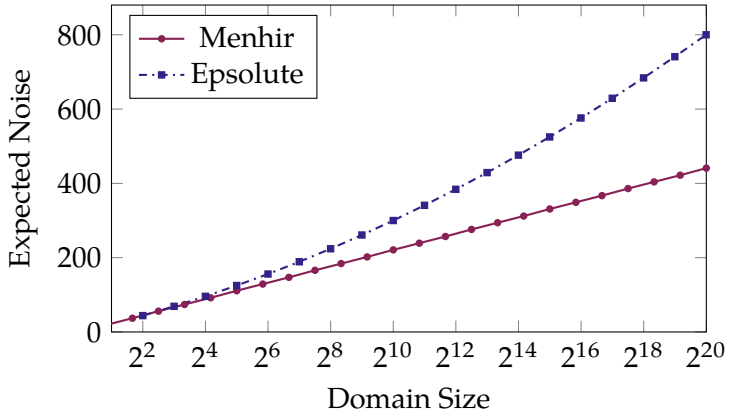


Figure 8.4: The size of the data domain D to the expected number of dummies needed for volume sanitation by Epsolute [55] and Menhir (lower is better). $\varepsilon = \ln 2, \delta = 2^{-20}$.

We prove the correctness and obliviousness of the ODB construction in Section B.2 in appendix B.

The General Data Protection Regulation (GDPR) [132] allows people whose data was processed to ask for it to be removed later. A `delete` functionality is therefore required. The hash h is computed from the data uploaded by the data subject or from identifiable information of the data subject. The latter allows the hash to be recreated if the person is no longer in its possession.

Volume Sanitation

To hide volume patterns, m values are retrieved from the database when processing a query. However, not all of these m data points are relevant to the final result for the query (as some are dummies). It is not possible to simply remove all dummies and then pass the array with all relevant values to a DP library, as this would again leak the volume of real data points. Instead, each DP function needs to process all m entries. When a dummy is processed, a dummy operation is made with the neutral element to this operation, e.g., adding a zero for summation. As the data distribution of each column c_i is different, the corresponding volume sanitizer S_i for this column needs to be initiated with suitable parameters. An overview of suitable sanitizers is provided by [255]. In our implementation, we rely on the Epsolute sanitizer [55]. However, by drawing noise from a truncated Laplace distribution, we achieve correctness with a smaller noise overhead compared to Epsolute.

The Epsolute volume sanitizer functions as follows. When initializing the database, the valid data range for each column is passed. For each column, a binary tree with D leaves is created, where D is the size of the (public, discretized) domain of the values in the respective column. Each node at level l in the sanitizer tree, starting with the leaves at $l = 0$, represents a range of 2^l possible values. Each node of this tree is associ-

ated with the number of database elements falling into the respective range. The volume for any queried range can then be computed by decomposing the query into power-of-2-sized ranges and summing the values associated with the corresponding tree nodes.

To provide DP, Epsolute perturbs the value of each node in the sanitizer tree with noise drawn from the Laplace distribution $\mathsf{Lap}(\alpha, \lambda)$. Here, $\lambda = 1/\varepsilon$ for point queries and $\lambda = \lceil \log_2(D) \rceil / \varepsilon$ for range queries. Here, α is chosen such that drawing D samples guarantees that all samples are positive with probability $1 - \beta$, for negligible β [55, Section 4.6].

We observe that we can guarantee correctness with probability 1 by using a truncated, shifted Laplace distribution $\mathsf{TSDLap}(t, \lambda)$ instead (see Section 8.1.3 in Chapter 2). This guarantees (ε, δ)-DP with non-zero probability δ. However, unlike β in Epsolute, the probability δ does not depend on the size of the domain D. Instead, it is only influenced by the $\log_2(D)$ ones drawn for any particular user. Figure 8.4 compares the expected number of dummies needed per node for Epsolute and Menhir for a fixed choice of $\varepsilon = \ln(2), \delta = 2^{-20}$. The number of nodes in a binary tree required by Epsolute to compute a_h is set to $2 \cdot D - 1$. The figure shows that for Epsolute, the number of dummies grows with $\mathcal{O}(log(D)^2)$ depending on the domain size D. For Menhir, the growth is slower with $\mathcal{O}(log(D))$. This effect becomes evident for larger domain sizes.

Output Sanitation

Before the analyzer can pose queries, data points that do not fulfill their quality requirements need to be filtered out. Since data can only be deleted and queries can only be posed afterward, there is no privacy risk for data subjects.

The data analyzer might be interested in a broad spectrum of information regarding the collected data. The problem is that even if the data analyzer is honest, they might be compromised or hacked without knowing. Therefore, to protect the privacy of data subjects, only differentially private aggregates are returned. Menhir provides the private aggregation functions `COUNT`, `SUM`, `MEAN`, `VARIANCE`, as well as the `MOST FREQUENT` and `LEAST FREQUENT` item by using the Report-Noisy-Max algorithm [108]. For a query q, a budget ϵ_q is used for the anonymization. The privacy loss can be quantified using the sequential composition theorem (see Section 8.1.3). For a total number n_Q of queries, this means the loss is bounded by $\sum_1^{n_Q} \epsilon_q^i$.

The sensitivity for each column is derived from the minimum and maximum values predefined by the attribute schema F. The database system maintains the privacy budget. Data analyzers posing queries can, however, specify how much of their budget they want to use for each query. If the budget is used up, no more queries can be processed.

While some works reset the privacy budget after a certain time [297], we refrain from this approach as the data in the database does not change after the querying phase starts.

Although the added noise changes the actual result, Bassily et al. [43] have shown that DP can improve the result of statistical analysis. This is because statistics aim to model a real distribution from observed samples and draw knowledge from this real distribution. DP algorithms can improve the generalization error which is introduced by the fact that only a limited number of samples are available.

Non-private databases cover a wide array of functions, such as providing SQL-like `GROUP-BY` functionality, multiple tables, and allowing different `JOIN` functions. Multiple tables can be easily implemented with our approach by using a new ODB for each table. However, `JOIN` functions have to ensure that volume patterns remain hidden. As this is a complex task in itself, we point to related work on this topic such as [190, 321]. Similarly, when realizing `GROUP-BY` functions, the number of groups must be either padded to the maximum or sanitized using DP. Wilson et al. [316] discuss how user contribution needs to be limited to provide DP guarantees for SQL-like `GROUP-BY` operations.

8.5 EVALUATION

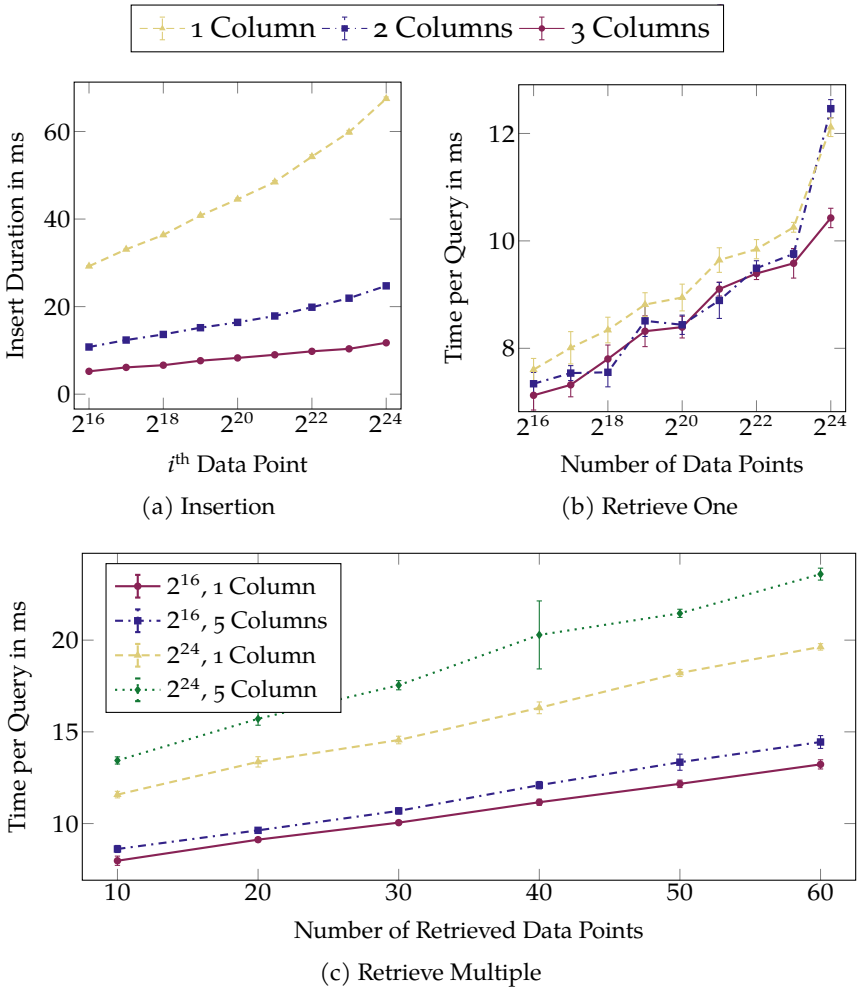
In this section, the performance and utility of the oblivious database is evaluated.

8.5.1 *Implementation and Measurements*

The Menhir oblivious database is implemented in C++. The implementation uses parts of the Epsolute [55] source code for hiding volume patterns. However, in Epsolute, the queried interval sometimes had to be increased to fit the buckets of the volume sanitizer tree. As a result, additional data points were retrieved due to this padding, which caused significant runtime overheads. Therefore, we adapted the code so that all leaves of the volume sanitizer tree are associated with exactly one element from the data domain instead of an interval. The Menhir source code calculates the volume sanitizers S_i once for each column c_i individually. The implementation of the oblivious database can handle integer and float values. When implementing the DP query function, leakage from floating point operations was considered [65].

As the authors of Oblix [224] did not make their DORAM implementation public, we opted for using the readily available Path-ORAM backend of Epsolute. As discussed in the threat model (see Section 8.3.2), data confidentiality is provided by the TEE. We, therefore, removed the AES encryption for ORAM blocks to improve runtime.

All evaluations were conducted on an AWS server with 121 GB RAM and 16 cores. The selected instance type `r6a.4xlarge` provides 3rd



(a) Insertion

(b) Retrieve One

(c) Retrieve Multiple

Figure 8.5: Performance of the ODB database for insertion, deletion, and query operations. (a) Insertion time of the i^{th} data point for ODBs with different numbers of columns. (b) Runtime for retrieving one data point from the ODB. (c) Runtime for retrieving a fixed number of data points from the ODB. Figure as in [6].

generation AMD EPYC processors (7003-series) for which the AMD SEV-SNP feature was enabled. Each data point in the following figures consists of at least ten measurements. Error bars represent the 95 % confidence intervals unless box plots are used. DP noise for volume and output sanitation was drawn from Laplace distributions.

8.5.2 *Performance*

Figure 8.5a shows how fast new elements can be inserted into Menhir's ODB. Using more columns corresponds to an increase in runtime for insertions, with a factor of 2.1 for two columns and a factor of 5.7 for five columns. This is due to the fact that for each additional column, a separate DOSM needs to be updated. Also, with an increasing amount

of data stored in the DOSM and the increasing size of the underlying ORAM, accesses take longer. We can see from the figure that insertion performs well even for a large number of data points and takes less than 10 ms for one column even when 2^{24} data points are already in the ODB.

The deletion operation also performs well despite the large amount of padding required to obfuscate the tree structure when deleting a node. For an ODB with 2^{24} data points, deletion of one element takes 45.63 ms for one column, respectively 264.73 ms for five columns.

For analyzers, it is important to know how fast their queries can be answered. Figure 8.5b shows how the query processing time is impacted by the number of data points in the ODB. The influence of the number of data points stored in the ODB on the runtime is logarithmic, while the impact of the number of columns is constant. The number of points retrieved from the ODB also affects query runtime. Figure 8.5c shows that the overhead of retrieving increasingly more data points from the ODB is a constant, independent of ORAM size and the number of columns.

Table 8.3: Comparison of the data protection guarantees and retrieval performance for multiple databases. The values for Oblix were taken from [224]. ORAM speed is the latency for ORAM operations for block size=64 Byte and ORAM size=10^5. Retrieval duration is given for ORAM size=2^{24}, with m being the number of data points retrieved. Table as in [6].

		Standard	Naive	Oblix	Menhir	Speed-up (Oblix/Mehir)
Access Pattern Protection		x	✓	✓	✓	
Volume Patterns Protection		x	✓	x	✓	
ORAM Speed		-	90.1 µs	125 µs	94.1 µs	1.3×
m=10	1 Column	0.107 ms	17.4 s	≈ 12.5 ms	11.6 ms	≈ 1×
	5 Columns	0.165 ms	24 s	-	13.4 ms	
m=60	1 Column	0.087 ms	732.4 s	≈ 25 ms	19.6 ms	1.3×
	5 Columns	0.082 ms	1065.4 s	-	23.6 ms	

8.5.3 *Comparison to Other Databases*

To compare against a naive baseline, we implemented a naive database that consists of a list of data points. To query this naive database, first, the number m' of data points to be returned is determined through a sanitized DP histogram. Then, an output array with dummy values is initiated with m' slots. Next, for every data point in the database, the naive algorithm iterates over each slot of the output array. If the

data point falls into the interval, the first dummy value encountered in the output array is overwritten. The complexity of this algorithm is $\mathcal{O}(n \cdot m')$.

In Table 8.3, we compare Menhir against a standard database without any privacy protection, the naive approach described above, and Oblix [224]. For the standard database without privacy, a MariaDB [219] SQL database was deployed on the AMD SEV-SNP server. Accesses were performed through a Python connector. As we can see from the table, Menhir's performance is in the same range as Oblix's, while it also provides volume sanitation guarantees. The speedup in runtime shown in the table is likely due to differences in implementation and used hardware. Menhir is faster than the naive approach by a factor of 1500 with the same protections and only between 108 times to 288 times slower than the approach without any protection. As we can see, unlike the naive approach, Menhir's runtime allows for a real-world deployment.

8.5.4 *Parallelization*

The linear increase in Figure 8.5c can be used to extrapolate the expected time it takes to retrieve a fixed number of data points. For an ODB with one column, 2^{24} data points and relying on an ORAM of the same size, it would take around 10.3 s to retrieve 2^{16} data points, respectively 15.1 s for five columns. For an ODB using an ORAM of size 2^{16}, retrieving the same amount of data requires only 6.4 s for one column, respectively 7.3 s for five columns. This insight can be used to improve the overall performance of Menhir for larger data sets. By storing data in multiple OSMs [30], which are accessed in parallel (each with a separate ORAM), data points can be retrieved faster and the worst-case runtime for large queries can be capped.

[30] *When talking about OSMs in the context of parallelization, it refers to the improved OSM construction consisting of a multi-index AVL tree that is stored in an ORAM.*

Parallelization in Menhir is achieved as follows. Multiple OSMs are associated with the database, each with its own ORAM of fixed size. New data points are always inserted into the newest OSM. A new OSM is created when the maximum capacity of the last one is reached. To delete a data point, all OSMs have to be checked. To calculate the response for a query, the data returned by all OSMs has to be combined.

Let us assume a data set of size 2^{24}. In the worst case, a query retrieves all data points in the database. Using the extrapolated runtime from earlier, it can be determined that each OSM should contain a maximum of 2^{16} data points. To hold the complete data set, 256 OSMs are required. An OSM of size 2^{16} requires 18.36 MB of RAM for one column, respectively 58.43 MB for five columns.

Parallelization itself also introduces an overhead to runtime due to caching effects. A query is also only as fast as the slowest thread. Table 8.4 shows the runtime for various data set sizes and OSM sizes to help determine the performance overhead introduced by accessing

Table 8.4: Overhead in ms of using multiple OSMs for two different OSM sizes. For each ODB, exactly 60 data points were retrieved in parallel. Table as in [6].

Data Set Size	DOSM Size 2^{16}	DOSM Size 2^{15}	Factor
2^{16}	9.62 ± 0.12	9.20 ± 0.07	1.05
2^{17}	10.19 ± 0.17	10.79 ± 0.13	1.06
2^{18}	11.93 ± 0.16	21.07 ± 0.41	1.77
2^{19}	23.73 ± 0.42	42.03 ± 0.31	1.77
2^{20}	44.73 ± 0.36	74.48 ± 0.93	1.66

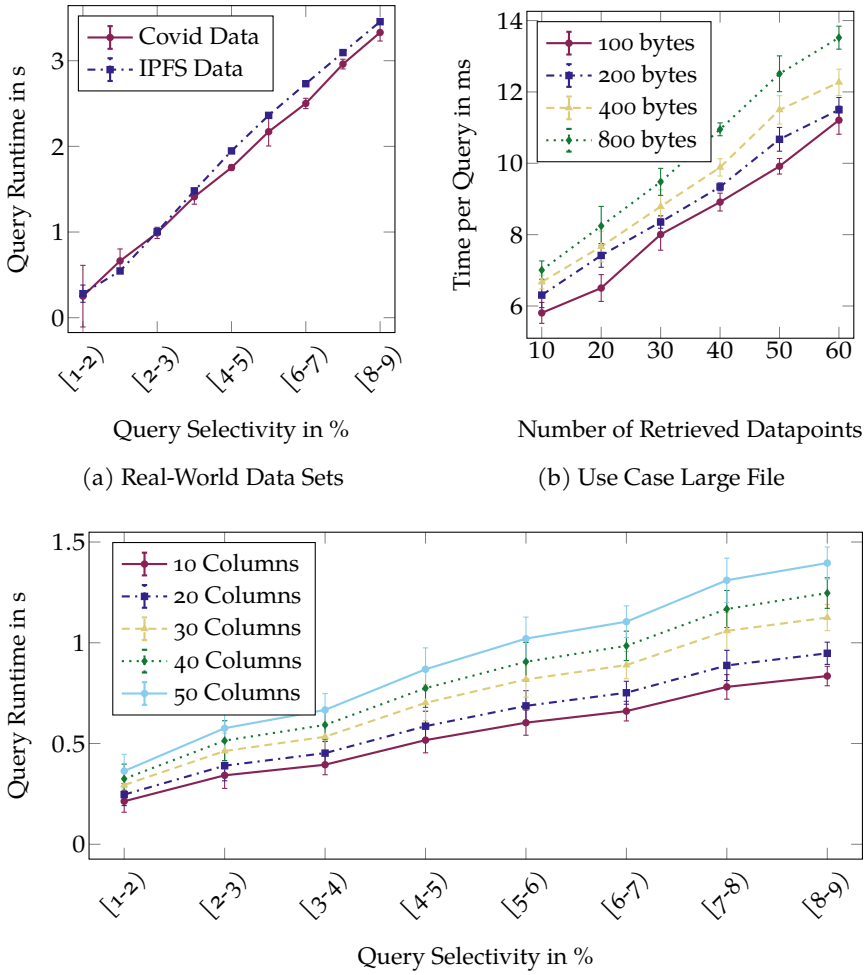
multiple OSMs in parallel. For each OSM, exactly 60 data points were retrieved in parallel. Note that the number of OSMs for different data set sizes depends on ORAM capacity. We can see that with an increasing total number of data points, the runtime increases despite parallelization. The drastic increase for 2^{18} data points suggests that this is a side effect caused by caching. Therefore, we repeated the measurements on a server with the same number of CPUs but a cache size 32 times as large. The results mirror the measurements from the AMD SEV-SNP server. However, the drastic increase for a DOSM size of 2^{15} is shifted to a data set size of 2^{20}. It is clear that the improvement of using multiple smaller OSMs is limited by the overhead of parallelization. Thus, despite better worst-case guarantees, the OSM size should be set with considerations for the average case.

To guarantee this performance, each OSM should be associated with one CPU core. It is not uncommon to have up to 64 cores even for consumer CPUs. In the case of a data set with 2^{24} data points, 4 machines with 16 cores each are sufficient to privately query the data set while guaranteeing one core per OSM. Offloading DOSMs to other machines does induce an overhead as intermediate results have to be communicated over the network. However, if the round delay between the central OSM and the satellite machines is low, parallelization improves the overall runtime.

8.5.5 *Real-World Data Sets and Use Cases*

In this section, we evaluate the utility of Menhir on real-world data sets and for different use cases. Many data sets used in related work were not relevant for evaluating a database that focuses on volume pattern sanitation, such as the key-value data sets used by Oblix [224]. Others were unavailable, such as the Big Data Benchmark data set used by Opaque [324]. For this reason, we evaluated Menhir on two other data sets [31]. As the TEE server used for measurements only has 16 CPUs, we

[31] *The data sets can be found at the author's Github repository* [11].



(a) Real-World Data Sets

(b) Use Case Large File

(c) Use Case Many Columns

Figure 8.6: Performance of Menhir for different use cases. (a) Query duration for different query selectivities on two different real-world datasets (dataset sizes= 2^{20}). (b) Time for retrieving a fixed number of data points, each associated with an unindexed file (dataset size=2^{16}). (c) Query selectivity to query duration for larger numbers of columns (dataset size=2^{16}). Figure as in [6].

limited the number of DOSMs to 16, allowing for a maximal data set size of 2^{20}.

Real-World Data Sets

To evaluate Menhir's performance on realistic data, we used a data set collected from user requests on the *Interplanetary File System* (IPFS), a Peer-to-Peer file storage system. The data was collected for research purposes and was provided by the authors of [41]. For this evaluation, all sensitive information, such as IP addresses and request IDs, was removed from the data and replaced with pseudonyms. The data set consists of one hour of captured IPFS traffic and contains 15,960,697 data points with eight attributes each. Figure 8.6a shows the runtime for queries with a selectivity of up to 9% (corresponding to 94,372 data points). Data is stored in ODBs of size 2^{16} and queried in parallel. In addition to the IPFS data set, another real-world data set with more columns was also tested. The Covid-19 data set [241] contains anonymized information on Mexican Covid-19 patients. It consists of 2^{20} data points and 21 columns.

Figure 8.6a shows how the query runtime changes for both data sets and different query selectivities. Despite capping the worst-case runtime, parallelization itself does introduce an overhead in the average case.

Many Columns

As shown in Section 8.5.2, the number of columns impacts the query runtime. Looking at the 20 most voted data sets from `kaggle.com` [175] in the categories "health" and "survey", the median number of columns is 32.5 with a maximum of 644 columns. To analyze the performance of Menhir on a large number of columns, Figure 8.6c shows the performance of Menhir for data sets with different numbers of columns and 2^{16} data points. The more columns of data are stored, the larger the performance penalty during querying becomes. With less than 1.5 s for a query on a table with 50 columns, the Menhir ODB is practical even in this case.

Searchable File Storage

The ODB allows associating each set of keys $[k_1, \cdots, k_C]$ with a value v. The size of this value is set when initializing the ODB. It can be used to associate a file with each tuple of keys or store additional data that does not need to be indexed itself. For all prior evaluations, the size of the value was set to zero. Figure 8.6b shows how different sizes impact query runtime for a data set with 2^{16} data points and one column. The figure makes clear that the overhead for having values of different sizes associated with each data point is constant. We can see that using

Menhir as a searchable file store with volume pattern sanitation is practical.

8.5.6 *Volume Sanitizer Overhead*

To hide the response volume of queries, Menhir uses volume pattern sanitizers. To estimate the number of data points that must be retrieved for queries, a sanitizer S_i is computed for each column c_i. The sanitizer S_i is a differentially private histogram for the data domain (see Section 8.4.2). The domain is computed based on the expected maximum and minimum values and the resolution of the data in column c_i.

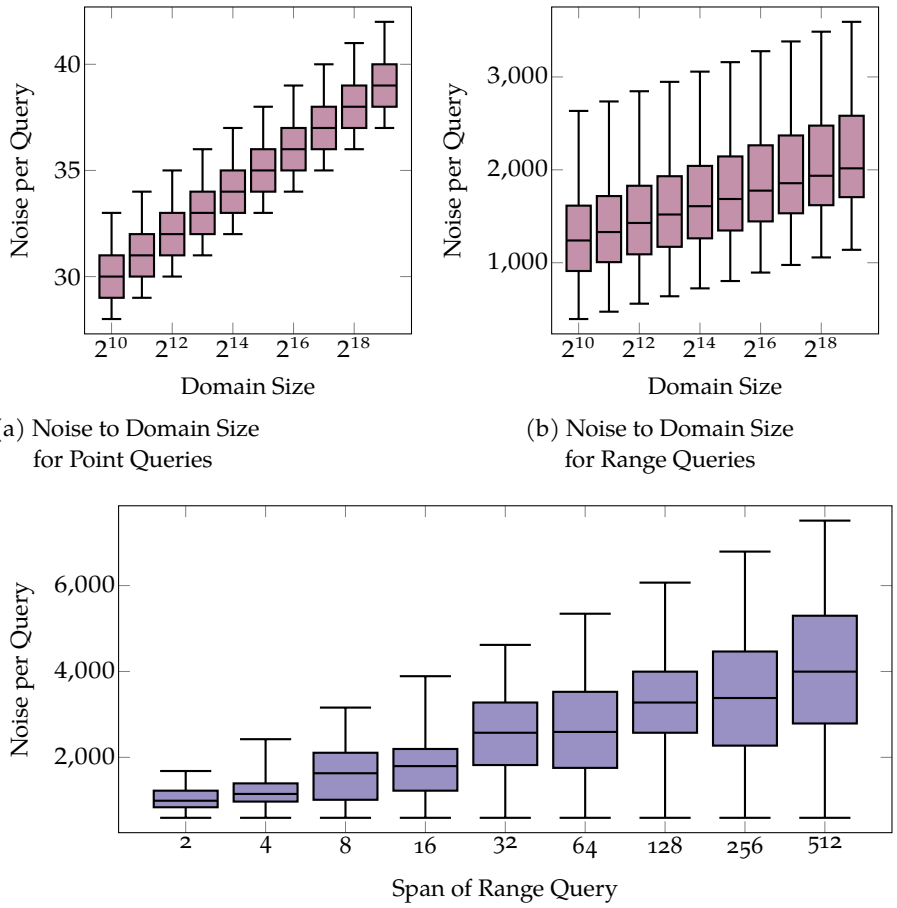
For point queries, the histogram's data structure is a flat array with as many buckets as there are elements in the domain. To deduce the noise required for a specific query, the corresponding bucket is checked to get the sanitized volume m.

For range queries, the data structure of the histogram is a tree. There are two approaches introduced by Epsolute [55] for volume sanitation in this case. The no-γ-method requires that for each OSM, an independent set of sanitizers is created. The γ-method, on the other hand, is optimized for a distributed setting and aims to keep the total noise added as low as possible. Here, even if multiple OSMs are used, only one sanitizer is required for each column.

Figure 8.7a and Figure 8.7b highlight the relationship between domain size and added noise for point queries and range queries. Here, the no-γ-method and normal Laplace noise were used to compute the amount of noise required for volume sanitation. All the queries used for these graphs have a selectivity of 1% and the span for range queries was fixed to 10. While the noise calculation for both point and range queries depends on the domain size, the influence is clearly visible for range queries but minimal for point queries. The progression mirrors the expected noise as shown in Figure 8.4. Differences in expected noise to measured noise can be explained by different parameterizations.

Figure 8.7c shows how much noise is added for range queries that cover different ranges but have the same selectivity when using the no-*gamma*-method and normal Laplace noise. With the increased range, more buckets in the sanitizer tree are required to cover the queried range. As each bucket adds DP noise to the query, the total noise per query increases. The plots show that the impact of the range of range queries on the amount of noise per query is larger than the impact of the domain size.

All three plots clearly emphasize the importance of setting well-suited parameters for the volume pattern sanitizers to best capture the (expected) data distribution of each column. This can be influenced by defining a data resolution. For example, if the expected minimal data resolution is 10, then buckets in the volume pattern sanitizer for any values in between are not necessary. This is especially relevant for float-



(a) Noise to Domain Size for Point Queries

(b) Noise to Domain Size for Range Queries

(c) Noise to Queried Range

Figure 8.7: The amount of differentially private noise required for volume sanitation. The number of data points is fixed to 2^{20} and the selectivity of queries is 1 %. (a) Noise applied to point queries depending on the domain size. Here, the span of range queries is fixed to 10. (b) Noise applied to range queries depending on the domain size. (c) Noise applied for range queries depending on the span of the query.Here, the domain size is fixed to 2^{15}. Figure as in [6].

ing point data as the volume pattern sanitizer only allows for a limited resolution of fixed size.

8.5.7 *Discussion*

As seen in Section 8.5.2, the runtime of each ODB is linear in the number of elements that fall into the queried interval. Parallelization allows limiting the worst-case runtime. This means Menhir is well suited if the expected selectivity of queries is low, for example, in heavily distributed or uniformly distributed data. If the database is first filtered by a column containing binary data, the worst-case performance is to be expected. Data resolution and how data is expected to be evaluated are relevant for the decision on how well-suited Menhir is for a specific use case.

Another potentially interesting use case for Menhir would be an interactive data collection setting where data can be queried while data collection continues. This is possible, yet not optimal, in a setting with a global privacy budget as the privacy budget might already be used up when new, relevant data points are inserted. Queries on these data points will not be answered in a setting with a global budget. Another issue for the interactive setting is the approach Menhir takes on volume pattern sanitation. Every time a query is posed after new data has been added, the volume pattern sanitizers would need to be recalculated. Hence, the privacy budget for sanitation in an interactive setting scales with the number of queries n_Q. We leave the optimization of this bound for an interactive query setting for future work.

8.6 CHAPTER SUMMARY

In this chapter, we presented Menhir, an oblivious database for TEEs such as Intel TDX and AMD SEV-SNP that protects against access pattern leakage and, unlike the previous works, also protects against volume pattern leakage. The database construction ensures that no trust needs to be placed on the infrastructure hosting the TEE, as it's options for learning sensitive information is limited. Additionally, it defends against malicious data analyzers trying to extract private data through crafted queries. Data providers can verify the integrity of Menhir before donating data by using remote attestation.

To arrive at the construction of Menhir, we first presented an attack against the state-of-the-art oblivious data structure Oblix [224] by using volume pattern leakage. To mitigate the volume pattern leakage, the underlying AVL tree construction was changed to allow the retrieval of fixed-size intervals from the tree. Building on the improved data structure an oblivious database was designed for which correctness and obliviousness were proven. By using a truncated Laplace function for generating DP noise for hiding volume patterns, the amount of noise is reduced and, therefore, also the performance overhead required. Our

evaluation showed that Menhir performs well even for many data points and multiple columns. Larger files can also be associated with each database row, while still retaining good query performance. This allows for a wide range of potential use cases of Menhir.

Part IV

EPILOGUE

CONCLUSION

This thesis focused on designs for preserving privacy during data collection and analysis that do not require data providers to trust in a central entity. The need for such research emerged during the Covid-19 pandemic as it highlighted the dilemmas faced by decision makers when balancing privacy and utility. The increase in new applications processing sensitive data for the common good sparked public discourse on the extent to which individuals are willing to sacrifice privacy to combat the pandemic. The legitimacy of utilizing such data, even beyond the pandemic, had to be re-negotiated time and time again. While some downplayed the risks of Covid-19 or even denied the existence of a pandemic altogether, others feared a rise in surveillance across all areas of life. Achieving the overall goal of stopping the spread of Covid-19 required persuading as many voluntary participants as possible. To reach those who were healthy but concerned about their data's current or future use, technical privacy guarantees served as tool for ensuring cooperation. In this context, privacy is not in conflict with utility. Instead, it amplifies it.

We contributed to the discourse in both the research community and the public by presenting an extensive overview of approaches to Digital Contact Tracing (DCT). The analysis of sub-problems and privacy issues provides a comprehensive collection of solutions for guiding the design of contact tracing applications. With proposals on contact tracing and super-spreader detection via presence tracing, we contributed to offering solutions that can provide various tracing functionalities without requiring mutual trust among the parties involved. This area of research prepares societies for potential future epidemics and pandemics while hoping for the best.

A positive effect of the pandemic and the widespread adoption of decentralized contact tracing is the increased public awareness of privacy-enhancing technologies. The pandemic's effect could serve as catalyst for the field, opening the door for utilization in new areas. As seen throughout this thesis, sensitive data accumulates during various steps of data processing and analysis. Only focusing on obvious identifiers when aiming to preserve privacy does not go far enough, as sources of privacy leakage and the potential for data misuse are manifold. For this reason, the focus of this thesis also encompassed the question of how new knowledge or statistical insight can be obtained from sensitive data. Here, we leveraged Trusted Execution Environments (TEEs) for provable privacy guarantees. This approach is in contrast to the a major dispute on the privacy of DCT as it opposes the division into centralized as non-private and distributed as inherently private by leveraging

remote attestation features of TEEs. Achieving the collection and evaluation of sensitive (medical) data without requiring mutual trust is a first step towards a more privacy-oriented world.

To facilitate research in the field, the code base of our privacy-preserving data analysis platform, Menhir, is open source. In addition to improving the functionality of Menhir, other questions and problems also need to be investigated further. While the data utilized in this thesis was mainly contact information or health data, anonymizing sequential data such as location traces or sequential measurements poses a distinct challenge. Such data could be utilized, for example, in future epidemics and pandemics by machine learning models in a privacy-preserving way to discover new infections.

Protocols that protect privacy need to be tailored to a specific use case to account for different sources of data breaches while remaining efficient. The challenges of the future, therefore, require customized solutions. However, the wheel does not need to be constantly reinvented, and such solutions can benefit from leveraging or extending existing ideas such as the one proposed in this thesis.

Part V

APPENDIX

A

APPENDIX: OVERVIEW OF DIGITAL CONTACT TRACING APPROACHES

This appendix contains multiple tables that provide an overview of Digital Contact Tracing (DCT) technologies and approaches.

Table A.1 compares multiple approaches that rely on the server for computing the risk of individual users. Here, approaches that do not protect the user's privacy by revealing the risk to the server are listed. Additionally, approaches are enumerated that do not reveal the user's risk to the server by relying on cryptographic protocols. Privacy and security issues are the focus of this table.

Table A.2 and Table A.3 give an overview of the privacy and security concerns of client-oriented approaches for risk calculation. Due to formatting, the three different categories, which are explained in detail in 3.4, are split over two tables.

Table A.4 summarizes the functionalities of different DCT designs, as discussed in Section 3.5. The focus lies on comparing how different DCT approaches authenticate uploads, verify encounters, and prove an infection risk to others. Additional functionalities, such as supporting international travel and uploading incomplete reports, are discussed by the approach.

Table A.1: Overview of contact tracing approaches with server-side risk calculation. (1): Gives pseudonym of diagnosed person and thereby time of encounter. (n.s.): Not specified how data from diagnosed users is collected. (n/a): Not applicable. (HA): Health Authority. Table as in [2].

	Name	Trust model for server	HA can track users	Results revealed to HA	Infected users can be deanonymized	Computation intensive	Defense against traffic analysis	Notes
Risk Revealed	TraceTogether/ BlueTrace [142]	T.	x	x				
	PePP-PT (NTK [247], Robert [161])	T.	x	x				
	Aarogya Setu [232]	T.	x	x				GPS+BLE
	PHyCT [170]	Semi-h.		x	(1)			
Risk Not Revealed	EPIC [29]	Semi-h.				x	n.s.	HE, Passively collected Wifi and Bluetooth Data
	HE-based TraceSecure [46]	Semi-h.				x	n/a	HE
	Berke et al. [49]	Semi-h.			(1)	x	n/a	GPS, MPC (PSI)
	Reichert et al. [5]	Semi-h./ M.			(1)	x	n.s.	GPS, MPC
	Demirag et al. [92]	Semi-h.				x	n.s.	MPC (PSI-CA)
	Epione [304]	Semi-h./ M.				x	n/a	HE+MPC (PSI-CA)
	CERTAIN [7]	Semi-h./ M.				x	n/a	MPC (circuit PSI)

Table A.2: Overview of contact tracing approaches with client-side risk calculation. (1): Gives pseudonym of diagnosed person and thereby time of encounter. (2): Gives time of encounter. (3): Cryptographic overhead on end devices. (4): Cryptographic and polling overhead on end devices. (HA): Health Authority. Table as in [2].

	Name	Trust model for server	HA can track users	Results revealed to HA	Infected users can be deanonymized	Computation intensive	Defense against traffic analysis	Notes
Broadcast	DP-3T [305]	Semi-h.			(1)	(3)	Cover traffic	
	GAEN [141]	Semi-h.			(1)	(3)		
	CONTAIN [153]	T.			(1)	Proto 1: (3)	Tor	
	East-Coast PACT (Rivest et al.) [266]	Semi-h.			(1)	(3)		
	West-Coast PACT (Chan et al.) [68]	Semi-h.			(1)	(3)		
	Hashomer [249]	Semi-h.			(1)	(3)		
Targeted	ConTra Corona [52]	T.		x	(2)		Tor	Requires non-colluding parties
	Covid-Watch [38]	Semi-h.			(1)	(3)		
	Pronto-C2 [40]	Semi-h.			(1)		Tor	Blockchain or server
	Whisper (decentralized) [212]	Semi-h.			(1)			Active BLE Protocol

Table A.3: Overview of contact tracing approaches with client-side risk calculation using messaging primitives. (1): Gives pseudonym of diagnosed person and thereby time of encounter. (2): Gives time of encounter. (3): Cryptographic overhead on end devices. (4): Cryptographic and polling overhead on end devices. (HA): Health Authority. Table as in [2].

	Name	Trust model for server	HA can track users	Results revealed to HA	Infected users can be deanonymized	Computation intensive	Defense against traffic analysis	Notes
Messaging	Cho et al. [75]	Semi-h.			(2)	(4)	Cover traffic and Tor	
	CAUDHT [1]	Semi-h.			(1)	(4)	Cover traffic and Tor	Uses DHT
	Ovid [4]	Semi-h.				(4)	Cover traffic and Tor	
	TraceSecure (messaging approach) [46]	Semi-h.	x	Partially	(2)		Cover traffic	Requires non-colluding parties
	Whisper (centralized) [212]	T.		Partially	(1)			Active BLE Protocol

Table A.4: A summary of additional functionalities of DCT systems. HA: Health Authority. Table as in [2].

Name	Authenticating Uploads	Verifying Encounters	Incomplete Reports	Proving Risk	International Travel
BlueTrace/ TraceTogether [142]	Tokens			HA knows	Supported
PePP-PT [161, 247]	Tokens			HA knows	Supported
PHyCT [170]				HA knows	
Epione [304]	Doctor collects data, tokens	Cryptographic hashes			
DP-3T [305]	Tokens	Locally		Verifying integrity of app	Supported
GAEN [141]	Tokens	Locally			Supported
CONTAIN [153]	Infection certificate	Locally			
East-Coast PACT [68]			Allowed		
West-Coast PACT [266]	Tokens	Locally			
Covid-Watch [38]					
Hashomer [249]		Locally	Allowed	Verification key	
Cho et al. [75]				Correct private key	
CAUDHT [1]	Tokens, blind signature	Locally		Correct private key	
Ovid [4]	Tokens, blind signature			Correct private key	
Pronto-C2 [40]	Tokens, blind signature	Locally	Allowed	Correct private key	
TraceSecure [46] (message-based)				Correct private key	
ConTra Corona [52]	Doctor collects data			Zero-knowledge proof	
Whisper [212]		Locally			

B

APPENDIX: MENHIR

B.1 ALGORITHMS

In this appendix, we provide and explain the pseudocode referenced in Section 8.4.1 and Section 8.4. We use lC and rC as shorthand for "left child" and "right child", the successor of node i is written as $node_i.succ$, and bT stands for "balance type".

Algorithm 2 DOSM.Insert(k, v)

$h \leftarrow \mathsf{hash}(k, v)$
$node_{new}, ptr_{new} \leftarrow \mathsf{AVLTreeNode}(k, v, h)$
$nodes \leftarrow [\ ]$
$ptr_{pre}, ptr_{parent} \leftarrow ptr_{dummy}$
$ptr_i \leftarrow ptr_{root}$
//Find insert location and predecessor
for $i \leftarrow 1$ to h_{max} **do**
  $node_i \leftarrow \mathsf{ORAM.Get}(ptr_i)$
  $nodes.append(node_i)$
  $left \leftarrow (k < k_i) \vee (k == k_i \wedge h < h_i)$
  $ptr_{pre} \leftarrow$ **if** not $left$ **then** ptr_i
  $ptr_{i+1} \leftarrow$ **if** $left$ **then** $node_i.lC$ **else** $node_i.rC$
  $isDummy \leftarrow (ptr_{i+1} == ptr_{dummy})$
  $ptr_{parent} \leftarrow$ **if** not $isDummy$ **then** ptr_i
  $ptr_i \leftarrow ptr_{i+1}$
//update parents (h_{max} write operations to ORAM)
$bT, ptr_{child}, ptr_{grandchild} \leftarrow \mathsf{UpdateParents}(nodes)$
//Update successor pointers
$node_{new} \leftarrow \mathsf{ORAM.Get}(ptr_{new})$
$node_{pre} \leftarrow \mathsf{ORAM.Get}(ptr_{pre})$
$isSmallest \leftarrow (ptr_{pre} == ptr_{dummy})$
$node_{pre}.succ \leftarrow$ **if** not $isSmallest$ **then** ptr_{new}
$node_{new}.succ \leftarrow$ **if** $isSmallest$ **then** ptr_{parent} **else** ptr_{pre}
$\mathsf{ORAM.Put}(\{node_{pre}, node_{new}\})$
//Insert node and rebalance tree
$\mathsf{Rebalance}(bT, ptr_{new}, ptr_{child}, ptr_{grandchild})$
return $\{h_i, ptr_{new}\}$

Algorithm 2 describes how a key-value pair $[k, v]$ can be inserted into a Doubly-Oblivious Sorted Multimap (DOSM) without leaking the structure of the tree. Conditions like "if c then r=a else r=b" can be

achieved without jumps through a single mathematical statement of the form:

$$r = c \cdot a + (\text{not } c) \cdot b$$

The simpler version of this condition is "if c then r=a". This can be implemented as follows:

$$r = c \cdot a + (\text{not } c) \cdot r$$

The sub-procedure Rebalance() relies on the insight in Section 8.4.1 that only a single or double rotation is sufficient to ensure that the AVL tree invariant is fulfilled after a new node is inserted. It performs 3 read and 3 write operations to Oblivious Random Access Memory (ORAM) independent of the fact whether any rebalancing is needed. Algorithm 3 shows the algorithm for the sub-procedure.

Algorithm 3 Rebalance($bT, ptr_{new}, ptr_{child}, ptr_{grandchild}$)

$node_{new} \leftarrow$ ORAM.Get(ptr_{new})
$node_c \leftarrow$ ORAM.Get(ptr_{child})
$node_g \leftarrow$ ORAM.Get($ptr_{grandchild}$)
//depending on balanceType bT these rotations are only dummy operations
Rotate($node_{new}$,$node_c$, bT)
Rotate($node_{new}$,$node_g$, bT)
ORAM.Put($\{node_{new}, node_c, node_g\}$)
return $\{h_i, ptr_{new}\}$

Algorithm 4 explains in detail how data is retrieved from the DOSM for an interval $[k_S, k_E]$ and a fixed number m. This m is selected to hide the volume of data in the queried interval. This algorithm returns a set of m nodes, each with all its associated data (keys and values). The function FindSmallesNodeInInterval() retrieves the smallest node for which the key k is larger or equal to k_S. If multiple nodes with the same key exist, they are ordered based on their hash. The function always makes h_{max} accesses to the ORAM.

Algorithm 4 DOSM.Find(k_S, k_E, m)

$ptr_{root} \leftarrow$ DOSM.root
$node_i \leftarrow$ FindSmallestNodeInInterval(ptr_{root}, k_S, k_E)
for $i \leftarrow 1$ to $m - 1$ **do**
    $R \leftarrow R \bigcup \{node_i\}$
    $node_{i+1} \leftarrow node_i.succ$
return R

In Section 8.4, an Oblivious Database (ODB) is built from the DOSM. Algorithm 5 explains how a data collector can query the ODB. During

data retrieval and computation, no volume patterns are leaked. The function takes as input a query consisting of an interval $[k_S, k_E]$, the column index c_w for which entries are retrieved (the column for the `WHERE`-clause), and the column index c_f for which the function f_j is applied on the retrieved entries. Function f_j is differentially private and is passed by the query through its index j. It uses the privacy budget ϵ_q for computing the differentially private aggregate. If the remaining global privacy budget is less than ϵ_q, the query cannot be answered.

Algorithm 5 ODB.Query$(k_S, k_E, c_w, c_f, j, \epsilon_q)$

$m \leftarrow S_{c_w}(k_S, k_E)$
$R \leftarrow$ ODB.Find(k_S, k_E, m, c_w)
$value \leftarrow f_j(c_f, \epsilon_q, R)$
return *value*

ODB.Query uses the find function from Algorithm 6 for retrieving m data points for the respective interval $[k_S, k_E]$ from the column c_w. For this purpose, it first finds the smallest node larger than or equal to k_S from the DOSM for column c_w. For each column, the ODB holds a pointer to the root node of the corresponding DOSM. Due to the use of the truncated Laplace function, all data points that fall into the interval $[k_S, k_E]$ are contained in the output of ODB.Find.

Algorithm 6 ODB.Find(k_S, k_E, m, c_w)

$ptr_{root} \leftarrow$ ODB.DOSMRoots$[c_w]$
$node_i \leftarrow$ FindSmallestNodeInInterval(ptr_{root}, k_S, k_E)
for $i \leftarrow 1$ to $m - 1$ **do**
    $R \leftarrow R \bigcup \{node_i\}$
    $node_{i+1} \leftarrow node_i.succ$
return R

B.2 CORRECTNESS AND OBLIVIOUSNESS

In the following, we prove the correctness and obliviousness of our ODB construction.

B.2.1 *Correctness*

Definition 3. (*Correctness*).
Let $x \in \{0,1\}^*$ represent the contents of a database table with multiple rows and columns. Function f is an operation on this table. A protocol π implementing f is correct *if the output of π is computationally indistinguishable from $f(x)$. In short, $output^{\pi}(x) \stackrel{c}{\equiv} f(x)$. This means for a negligible function μ it holds that*

$$Pr[output^{\pi}(x) = f(x)] \geq 1 - \mu.$$

We first introduce some notation. As mentioned before, a query $q \in Q$ is a tuple $q = (k_S, k_E, c_w, c_f, j, \epsilon_q)$. Let V_q be the set containing all entries from column c_f for which the corresponding entry in column c_w fulfills the query condition $k_S \leq c_w \leq k_E$. For proper privacy protection, we require that the sensitivity of f_j is set correctly for the data type in column c_f.

Theorem 1. *Following Definition 3, the ODB scheme in Section 8.4 is correct for the functions* ODB.Init, ODB.Insert *and* ODB.Delete.

Proof. This follows from the correctness of the oblivious data structure framework of Wang et al. [314] and the plaintext AVL tree construction which the DOSM builds on. □

The correctness of the ODB.Query function boils down to the correctness of the ODB.Find function and the correctness of the function f_j evaluation. First, we discuss the correctness of the ODB.Find function.

Theorem 2. ODB.Find *correctly returns all elements in an interval given by a query $q \in Q$ if $m \geq |V_q|$.*

Proof. The ODB.Find function makes a call to the DOSM.Find function for finding the elements in a given interval. Therefore, the correctness of the ODB.Find function follows from the correctness of the DOSM.Find function for a particular column c_f. □

Theorem 3. *The* ODB.Query *function correctly computes the function f on the required entries.*

Proof. As mentioned earlier, the correctness of the ODB.Query function depends on the correctness of the ODB.Find function and the computation of function f. Theorem 2 proves the correctness of ODB.Find. To show that the query function f is computed correctly, it suffices to show that (a) all required elements are included in the aggregation and (b) the dummies added through the volume sanitizer do not change the output. Claim (a) follows directly from the fact that our volume sanitizer always adds positive noise through the truncated Laplace mechanism (see Section 8.4.2). For this reason, the constraint $m \geq |V_q|$ always holds. This is in contrast to Epsolute [55], which fails with a (negligible) probability β. Claim (b) can be ensured to hold by using the neutral element of the respective aggregation function as the value for dummies. As a result, computation on dummies don't change the output.

□

Remark: A malicious adversary can insert data points in the ODB and delete these. This allows them to alter the result of all queries and skew the data analysis. However, this is a general risk of crowdsourcing campaigns, and in particular, it does not depend on or reveal any user's input. Sybil attacks can be made more resource intensive for attackers

by requiring data providers to identify themselves. To preserve privacy, the identification process can be implemented through anonymous authentication schemes. However, such schemes are not the focus of this work, so we point the reader to the relevant work on this topic, such as [62].

B.2.2 *Obliviousness*

Theorem 4. *The* ODB.Init, ODB.Insert, *and* ODB.Delete *operations of the ODB scheme are oblivious with a leakage function* $\mathcal{L} = ((op_1, c_1) \cdots, (op_M, c_M))$ *according to Definition 1 on page 131. The leakage function leaks only the operation type* op_i*, the accessed column* c_i*, and the total number of operations* M*, but nothing else.*

Proof. First, observe that ODB.Init, ODB.Insert, and ODB.Delete make the same number and types of ORAM accesses for two function calls even if different data is provided. It then follows immediately from the security of the underlying ORAM scheme [292] that the memory addresses produced are indistinguishable for any two function calls. We can, therefore, define the simulator $\mathcal{S}$ that takes the sequence of operations and then runs the corresponding algorithms on a dummy index-value pair (say, $(0, 0)$). □

Without additional perturbation, the volume of data used for answering a query can be used to reconstruct a database [181, 194]. Algorithms for oblivious databases that do not pay attention to this side channel, such as Oblix [224], end up leaking this information (see Section 8.4.1). To obliviously answer queries, one could process the whole database for each query. However, this is not very efficient. Therefore, we weaken the definition by allowing additional leakage per operation but requiring that this leakage be differentially private, in accordance to with Definition 2 for Differential Privacy (DP).

Definition 4. (*Obliviousness with DP volume leakage*).
Let m_i *be the volume of data processed for an operation* op_i *on column* c_i *with arguments* arg_i*. A data structure* $\mathcal{D}$ *is oblivious with DP volume leakage, if there exists a polynomial time simulator* $\mathcal{S}$*, such that for any polynomial-length sequence of data structure operations* $\overrightarrow{ops} = ((op_1, c_1, args_1, m_1), ..., (op_M, c_M, args_M, m_M))$ *and leakage* $\mathcal{L}(\overrightarrow{ops}) = ((op_1, c_1, m_1), \cdots, (op_M, c_M, m_M))$ *it holds that*

$$addresses_{\mathcal{D}}(\overrightarrow{ops}) \stackrel{c}{\equiv} \mathcal{S}(\mathcal{L}(\overrightarrow{ops}))$$

and each m_i *provides* (ε, δ)*-differential privacy with respect to individual database items.*

Theorem 5. *For an ODB with* (ε, δ)*-DP volume sanitation, a sequence of* M ODB.Query *operations is oblivious with* $(M \cdot \epsilon, M \cdot \delta)$*-DP volume leakage.*

Proof. We start by defining the simulator $\mathcal{S}$ that takes as input the leakage $\mathcal{L}$ containing the operations op_i, sanitized volumes m_i, and column indices c_i, then calls ODB.Query on c_i and a dummy key, replacing m in the first line by m_i.

Since FindSmallestNodeInInterval() makes the same number of ORAM queries independent of the arguments, the resulting addresses will be indistinguishable from the real implementation of ODB.Query.

It remains to be shown that each m_i is differentially private. As described in Section 8.4.2, the volume sanitizer uses a binary tree, where each database item is counted exactly once per level. If we consider two neighboring databases that differ in exactly one item x, there will therefore be exactly $h = \lceil \log_2(D) \rceil$ nodes that x contributes to. Since each node has noise drawn from $\mathsf{TSDLap}(t, h/\varepsilon)$, revealing a single node's value is $(\varepsilon/h, \delta/h)$-DP for appropriately chosen t (see Section 8.1.3). By basic composition, the revealing all h nodes is, therefore, (ε, δ)-DP. The claim follows through basic composition across M queries.

□

Part VI

BIBLIOGRAPHY

BIBLIOGRAPHY

PEER-REVIEWED PUBLICATIONS BY THE AUTHOR

[1] Samuel Brack, Leonie Reichert, and Björn Scheuermann. "CAU-DHT: Decentralized Contact Tracing Using a DHT and Blind Signatures." In: *Proceedings of the 45th IEEE Conference on Local Computer Networks, LCN 2020, Sydney, Australia, November 16-19, 2020*. IEEE, 2020, pp. 337–340. DOI: 10.1109/LCN48667.2020.9314850 (cit. on pp. 8, 28, 33, 50, 51, 97, 165, 166).

[2] Leonie Reichert, Samuel Brack, and Björn Scheuermann. "A Survey of Automatic Contact Tracing Approaches Using Bluetooth Low Energy." In: *ACM Transactions on Computing for Healthcare* 2.2 (2021), pp. 1–33. DOI: 10.1145/3444847 (cit. on pp. 8, 9, 14, 17, 19, 21, 25, 163–166).

[3] Leonie Reichert, Samuel Brack, and Björn Scheuermann. "Lighthouses: A Warning System for Super-Spreader Events." In: *Proceedings of the IEEE ICC Workshop on Communication, IoT, and AI Technologies to Counter Covid-19, COVI-COM 2021, Online, June 14-23, 2021*. Workshop on Communication, IoT, and AI Technologies to Counter Covid-19 (COVI-COM). IEEE, 2021, pp. 1–6. DOI: 10.1109/ICCWORKSHOPS50388.2021.9473758 (cit. on pp. 89, 94, 100).

[4] Leonie Reichert, Samuel Brack, and Björn Scheuermann. "Ovid: Message-Based Automatic Contact Tracing." In: *Proceedings of the NDSS Workshop on Secure IT Technologies against Covid-19, CoronaDef 2021, Online, February 21, 2021*. Innovative Secure IT Technologies against Covid-19 (CoronaDef) Workshop. Internet Society, 2021, pp. 1–8. URL: https://www.ndss-symposium.org/ndss-program/coronadef-2021/ (cit. on pp. 8, 54, 60, 165, 166).

[5] Leonie Reichert, Samuel Brack, and Björn Scheuermann. "Poster: Privacy-Preserving Contact Tracing of Covid-19 Patients." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2020 - Poster Session, Online, May 18-20, 2020*. IEEE, 2020, pp. 1–2. URL: https://www.ieee-security.org/TC/SP2020/program-posters.html (cit. on pp. 9, 16, 67, 70, 163).

[6] Leonie Reichert, Gowri R Chandran, Phillipp Schoppmann, Thomas Schneider, and Björn Scheuermann. "Menhir: An Oblivious Database with Protection against Access and Volume Pattern Leakage." In: *Proceedings of the ACM Asia Conference on*

Computer and Communications Security, ASIA CCS 2024, Singapore, Singapore, July 1–5, 2024. New York, NY, USA: ACM. DOI: 10.1145/3634737.3657005 (cit. on pp. 127, 135, 136, 140, 141, 144, 148, 149, 151, 152, 155).

[7] Leonie Reichert, Marcel Pazelt, and Björn Scheuermann. "Circuit-Based PSI for Covid-19 Risk Scoring." In: *Proceedings of the IEEE International Performance, Computing, and Communications Conference, IPCCC 2021, Austin, TX, USA, October 29-31, 2021*. IEEE, 2021, pp. 1–8. DOI: 10.1109/IPCCC51483.2021.9679360 (cit. on pp. 9, 72, 81–83, 163).

[8] Leonie Reichert and Björn Scheuermann. "An Analysis of Requirements and Privacy Threats in Mobile Data Donations." In: *Proceedings of the IEEE EuroS&P International Workshop on Privacy Engineering, IWPE 2024, Delft, Netherlands, July 3-7, 2023*. International Workshop on Privacy Engineering (IWPE). IEEE, 2023, pp. 84–93. DOI: 10.1109/EUROSPW59978.2023.00015 (cit. on pp. 108, 110, 117–119, 122).

[9] Phillipp Schoppmann, Adrià Gascón, Leonie Reichert, and Mariana Raykova. "Distributed Vector-OLE: Improved Constructions and Implementation." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. Leonie Reichert contributed to this work within the scope of her Master thesis. ACM, 2019, pp. 1055–1072. DOI: 10.1145/3319535.3363228 (cit. on p. 87).

OTHER PUBLICATIONS BY THE AUTHOR

[10] Samuel Brack, Jeanette Hofmann, Leonie Reichert, and Björn Scheuermann. *Tracing-Technologien: Die Corona-App Ihres Vertrauens*. Accessed: 2024-01-10. 2020. URL: https://netzpolitik.org/2020/die-corona-app-ihres-vertrauens/ (cit. on p. 4).

[11] Leonie Reichert. *Menhir*. Accessed: 2024-03-12. 2024. URL: https://github.com/ReichertL/Menhir (cit. on p. 151).

OTHER PUBLICATIONS

[12] DP-3T Team. *Best Practices Operational Security for Proximity Tracing*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Best%20Practices%20for%20Operation%20Security%20in%20Proximity%20Tracing.pdf (cit. on pp. 36–38).

[13] DP-3T Team. *BLE Measurements*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/bt-measurements (cit. on p. 15).

[14] DP-3T Team. *Decentralized Proximity Tracing Interoperability Specification - Release 0.1 (Draft)*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Interoperability%20Decentralized%20Proximity%20Tracing%20Specification%20(Preview).pdf (cit. on p. 30).

[15] DP-3T Team. *DP-3T Exposure Score Calculation - Summary*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Exposure%20Score%20Calculation.pdf (cit. on p. 26).

[16] DP-3T Team. *Failure to Rotate RPI and MAC Addresses*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/bt-measurements/blob/master/linkability.md (cit. on p. 34).

[17] DP-3T Team. *Privacy and Security Attacks on Digital Proximity Tracing Systems*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/documents/blob/master/Security%20analysis/Privacy%20and%20Security%20Attacks%20on%20Digital%20Proximity%20Tracing%20Systems.pdf (cit. on p. 32).

[18] DP-3T Team. *Secure Upload Authorisation for Digital Proximity Tracing*. Accessed: 2024-01-11. 2020. URL: https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Upload%20Authorisation%20Analysis%20and%20Guidelines.pdf (cit. on pp. 28, 37, 40).

[19] Aargauer Zeitung. *Kommission will keine Pflicht für Nutzung von Contact-Tracing-App*. Aargauer Zeitung. Accessed: 2024-01-11. 2020. URL: https://www.aargauerzeitung.ch/news-service/inland-schweiz/kommission-will-keine-pflicht-fuer-nutzung-von-contact-tracing-app-ld.1214894 (cit. on pp. 3, 9, 39).

[20] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. "A Survey on Homomorphic Encryption Schemes: Theory and Implementation." In: *ACM Computing Surveys* 51.4 (2018), pp. 1–35. DOI: 10.1145/3214303 (cit. on p. 134).

[21] Adil Ahmad et al. "OBFUSCURO: A Commodity Obfuscation Engine on Intel SGX." In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. URL: https://www.ndss-symposium.org/ndss-paper/obfuscuro-a-commodity-obfuscation-engine-on-intel-sgx/ (cit. on p. 134).

[22] Esma Aïmeur, Oluwa Lawani, and Kimiz Dalkir. "When Changing the Look of Privacy Policies Affects User Trust: An Experimental Study." In: *Computers in Human Behavior* 58 (2016), pp. 368–379. DOI: 10.1016/J.CHB.2015.11.014 (cit. on pp. 114, 120).

[23] Ahmet Aktay et al. "Google Covid-19 Community Mobility Reports: Anonymization Process Description (version 1.0)." In: *Computing Research Repository (CoRR)* abs/2004.04145 (2020). URL: https://arxiv.org/abs/2004.04145 (cit. on pp. 127, 128).

[24] Mimonah Al Qathrady, Ahmed Helmy, and Khalid Almuzaini. "Infection Tracing in Smart Hospitals." In: *Proceedings of the 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2016, New York, NY, USA, October 17-19, 2016*. IEEE Computer Society, 2016, pp. 1–8. DOI: 10.1109/WIMOB.2016.7763193 (cit. on p. 11).

[25] Reed Albergotti and Drew Harwell. *Apple and Google are Building a Virus-Tracking System. Health Officials Say it Will Be Practically Useless.* Washington Post. Accessed: 2024-02-09. 2020. URL: https://www.washingtonpost.com/technology/2020/05/15/app-apple-google-virus/ (cit. on p. 89).

[26] Bakheet Aljedaani and M Ali Babar. "Challenges with Developing Secure Mobile Health Applications: Systematic Review." In: *JMIR mHealth and uHealth* 9.6 (2021). DOI: https://doi.org/10.2196%2F15654 (cit. on p. 109).

[27] Titan Alon, Minki Kim, David Lagakos, and Mitchell VanVuren. *How Should Policy Responses to the Covid-19 Pandemic Differ in the Developing World?* Tech. rep. Accessed: 2024-01-26. National Bureau of Economic Research, 2020. URL: https://www.nber.org/papers/w27273 (cit. on p. 127).

[28] Samuel Altmann et al. "Acceptability of App-Based Contact Tracing for Covid-19: Cross-Country Survey Study." In: *JMIR mHealth and uHealth* 8.8 (2020). DOI: 10.2196/19857 (cit. on p. 39).

[29] Thamer Altuwaiyan, Mohammad Hadian, and Xiaohui Liang. "EPIC: Efficient Privacy-Preserving Contact Tracing for Infection Detection." In: *Proceedings of the IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20-24, 2018*. IEEE, 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422886 (cit. on pp. 11, 16, 18, 20, 65, 163).

[30] AMD. *AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More*. Accessed: 2024-01-29. 2020. URL: https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf (cit. on pp. 127, 130, 137).

[31] Moran Amit et al. "Mass-Surveillance Technologies to Fight Coronavirus Spread: The Case of Israel." In: *Nature Medicine* 26.8 (2020), pp. 1167–1169. DOI: 10.1038/s41591-020-0927-z (cit. on p. 67).

[32] Android Developers. *Android NDK*. Accessed: 2024-01-12. 2024. URL: https://developer.android.com/ndk (cit. on p. 80).

[33] Sebastian Angel and Srinath T. V. Setty. "Unobservable Communication Over Fully Untrusted Infrastructure." In: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*. USENIX Association, 2016, pp. 551–569. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/angel (cit. on p. 124).

[34] Apple. *About iOS 13 Updates*. Accessed: 2024-03-24. 2024. URL: https://support.apple.com/en-us/118392 (cit. on p. 23).

[35] Apple. *ENExposureConfiguration*. Accessed: 2024-01-11. 2020. URL: https://developer.apple.com/documentation/exposurenotification/enexposureconfiguration (cit. on p. 77).

[36] Apple. *ResearchKit*. Accessed: 2024-02-06. 2024. URL: https://developer.apple.com/design/human-interface-guidelines/researchkit (cit. on p. 108).

[37] Aradhana Aravindan and Sankalp Phartiyal. *Bluetooth Phone Apps for Tracking Covid-19 Show Modest Early Results*. Reuters. Accessed: 2024-01-11. 2020. URL: https://www.reuters.com/article/us-health-coronavirus-apps/bluetooth-phone-apps-for-tracking-covid-19-show-modest-early-results-idUSKCN2232A0/ (cit. on p. 20).

[38] Sydney von Arx et al. *Slowing the Spread of Infectious Diseases Using Crowdsourced Data*. Accessed: 2024-03-04. 2020. URL: https://www.wehealth.org/covidwatchorg/blog/covid-watch-whitepaper-using-crowdsourced-data-to-slow-virus-spread (cit. on pp. 18, 24, 38, 48, 164, 166).

[39] World Medical Association et al. "World Medical Association Declaration of Helsinki: Ethical Principles for Medical Research Involving Human Subjects." In: *Journal of the American Medical Association (Jama)* 310.20 (2013), pp. 2191–2194. DOI: 10.1001/jama.2013.281053 (cit. on p. 118).

[40] Gennaro Avitabile, Vincenzo Botta, Vincenzo Iovino, and Ivan Visconti. "Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System." In: *IACR Cryptology ePrint Archive* (2020), p. 493. URL: https://eprint.iacr.org/2020/493 (cit. on pp. 18, 24, 28, 29, 33, 37, 47, 48, 164, 166).

[41] Leonhard Balduf, Sebastian A. Henningsen, Martin Florian, Sebastian Rust, and Björn Scheuermann. "Monitoring Data Requests in Decentralized Data Storage Systems: A Case Study of IPFS." In: *Proceedings of the 42nd IEEE International Conference on Distributed Computing Systems, ICDCS 2022, Bologna, Italy, July 10-13, 2022*. IEEE, 2022, pp. 658–668. DOI: 10.1109/ICDCS54860.2022.00069 (cit. on p. 153).

[42] Tala Ballouz et al. "Individual-Level Evaluation of the Exposure Notification Cascade in the SwissCovid Digital Proximity Tracing App: Observational Study." In: *JMIR Public Health and Surveillance* 8.5 (2022). DOI: 10.2196/35653 (cit. on p. 41).

[43] Raef Bassily et al. "Algorithmic Stability for Adaptive Data Analysis." In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. ACM, 2016, pp. 1046–1059. DOI: 10.1145/2897518.2897566 (cit. on p. 147).

[44] Lars Baumgärtner et al. "Mind the GAP: Security & Privacy Risks of Contact Tracing Apps." In: *Proceedings of the 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020, Guangzhou, China, December 29, 2020 - January 1, 2021*. IEEE, 2020, pp. 458–467. DOI: 10.1109/TRUSTCOM50675.2020.00069 (cit. on pp. 33, 34, 120).

[45] Martin Z Bazant and John WM Bush. "A Guideline to Limit Indoor Airborne Transmission of Covid-19." In: *Proceedings of the National Academy of Sciences* 118.17 (2021). DOI: 10.1073/pnas.2018995118 (cit. on pp. 100, 101).

[46] James Bell, David Butler, Chris Hicks, and Jon Crowcroft. "TraceSecure: Towards Privacy Preserving Contact Tracing." In: *Computing Research Repository (CoRR)* abs/2004.04059 (2020). URL: https://arxiv.org/abs/2004.04059 (cit. on pp. 18, 20, 25, 37, 49, 66, 163, 165, 166).

[47] James Bell et al. "Distributed, Private, Sparse Histograms in the Two-Server Model." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angles, CA, USA, November 7 - 11, 2022*. ACM, 2022, pp. 307–321. DOI: 10.1145/3548606.3559383 (cit. on p. 132).

[48] Britt Elise Bente et al. "The Dutch Covid-19 Contact Tracing App (the CoronaMelder): Usability Study." In: *JMIR Formative Research* 5.3 (2021). DOI: 10.2196/27882 (cit. on p. 40).

[49] Alex Berke et al. "Assessing Disease Exposure Risk with Location Histories and Protecting Privacy: A Cryptographic Approach in Response to a Global Pandemic." In: *Computing Research Repository (CoRR)* abs/2003.14412 (2020). URL: https://arxiv.org/abs/2003.14412 (cit. on pp. 18, 20, 67, 163).

[50] Berliner Zeitung. *Raus aus dem Lockdown - Corona-Warn-App steht zum Download bereit, aber es gibt noch Forderungen*. Accessed: 2024-01-11. 2020. URL: https://www.berliner-zeitung.de/zukunft-technologie/corona-warn-app-starttermin-am-dienstag-steht-aber-es-gibt-noch-forderungen-li.87669 (cit. on pp. 39, 42).

[51] Stefano Bertuletti, Andrea Cereatti, Ugo Della Della, Michele Caldara, and Michael Galizzi. "Indoor Distance Estimated from Bluetooth Low Energy Signal Strength: Comparison of Regression Models." In: *Proceedings of the IEEE Sensors Applications Symposium, SAS 2016, Catania, Italy, April 20-22, 2016*. IEEE, 2016, pp. 1–5. DOI: 10.1109/SAS.2016.7479899 (cit. on p. 14).

[52] Wasilij Beskorovajnov et al. "ConTra Corona: Contact Tracing against the Coronavirus by Bridging the Centralized-Decentralized Divide for Stronger Privacy." In: *Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2021, Singapore, December 6-10, 2021*. Vol. 13091. Lecture Notes in Computer Science. Springer, 2021, pp. 665–695. DOI: 10.1007/978-3-030-92075-3_23 (cit. on pp. 18, 24, 25, 27, 30, 34, 36, 38, 47, 164, 166).

[53] Bluetooth SIG. *2020 Bluetooth Market Update*. Accessed: 2024-01-11. 2020. URL: https://www.bluetooth.com/bluetooth-resources/2020-bmu/ (cit. on p. 14).

[54] Bluetooth SIG. *Core Specification 4.0*. Accessed: 2024-02-06. 2023. URL: https://www.bluetooth.com/specifications/specs/core-specification-4-0/ (cit. on pp. 14, 15).

[55] Dmytro Bogatov, Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. "εpsolute: Efficiently Querying Databases While Providing Differential Privacy." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications*

Security, CCS 2021, Online, November 15 - 19, 2021. ACM, 2021, pp. 2262–2276. DOI: 10.1145/3460120.3484786 (cit. on pp. 70, 128, 133, 136, 139, 145–147, 154, 170).

[56] Simone Brienza et al. "A Survey on Energy Efficiency in P2P Systems: File Distribution, Content Streaming, and Epidemics." In: *ACM Computing Surveys* 48.3 (2016), pp. 1–37. DOI: 10.1145/2835374 (cit. on p. 53).

[57] Alexander Brunkow. "Implementation and Evaluation of a Super-Spreader Warning System." Bachelor Thesis. Humboldt University Berlin, 2021 (cit. on pp. 99, 103).

[58] Fabian Buder et al. *Adoption Rates for Contact Tracing App Configurations in Germany*. Accessed: 2024-01-11. 2020. URL: https://www.nim.org/en/publications/detail/research-report-adoption-rates-for-contact-tracing-app (cit. on pp. 39, 92).

[59] Angelique Burdinski, Dirk Brockmann, and Benjamin Frank Maier. "Understanding the Impact of Digital Contact Tracing during the Covid-19 Pandemic." In: *PLOS Digital Health* 1.12 (2022). DOI: 10.1371/journal.pdig.0000149 (cit. on pp. 40, 41).

[60] Matt Burgess. *Coronavirus Contact Tracing Apps Were Meant to Save Us. They won't.* Wired. Accessed: 2024-01-11. 2020. URL: https://www.wired.co.uk/article/contact-tracing-apps-coronavirus (cit. on pp. 9, 20).

[61] Garance Burke, Josef Federman, Huizhong Wu, Krutika Pathi, and Rod Mcguirk. *Police Seize on Covid-19 Tech to Expand Global Surveillance*. AP News. Accessed: 2024-02-15. 2022. URL: https://apnews.com/article/technology-police-government-surveillance-covid-19-3f3f348d176bc7152a8cb2dbab2e4cc4 (cit. on p. 43).

[62] Jan Camenisch and Anna Lysyanskaya. "Signature Schemes and Anonymous Credentials from Bilinear Maps." In: *Proceedings of the 24th Annual International Cryptology Conference, CRYPTO 2004, Santa Barbara, California, USA, August 15-19, 2004*. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 56–72. DOI: 10.1007/978-3-540-28628-8_4 (cit. on p. 171).

[63] Darlan S Candido et al. "Evolution and Epidemic Spread of SARS-CoV-2 in Brazil." In: *Science* 369.6508 (2020), pp. 1255–1260. DOI: 10.1126/science.abd2161 (cit. on p. 127).

[64] Elinor Carmi et al. *Joint Statement on Contact Tracing: Date 19th April 2020*. Accessed: 2024-01-11. 2020. URL: https://www.esat.kuleuven.be/cosic/sites/contact-tracing-joint-statement/ (cit. on p. 20).

[65] Sílvia Casacuberta, Michael Shoemate, Salil P. Vadhan, and Connor Wagaman. "Widespread Underestimation of Sensitivity in Differentially Private Libraries and How to Fix It." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*. ACM, 2022, pp. 471–484. DOI: 10.1145/3548606.3560708 (cit. on p. 147).

[66] Giulia Cencetti et al. "Digital Proximity Tracing on Empirical Contact Networks for Pandemic Control." In: *Nature Communications* 12.1 (2021), p. 1655. DOI: 10.1038/s41467-021-21809-w (cit. on p. 41).

[67] Centers for Disease Control and Prevention, Department of Health and Human Services USA. *Information for Health Departments on Reporting Cases of Covid-19*. Accessed: 2024-05-23. 2023. URL: https://www.cdc.gov/coronavirus/2019-ncov/php/reporting-pui.html (cit. on pp. 10, 12).

[68] Justin Chan et al. "PACT: Privacy-Sensitive Protocols and Mechanisms for Mobile Contact Tracing." In: *IEEE Data Engineering Bulletin* 43.2 (2020), pp. 15–35. URL: http://sites.computer.org/debull/A20june/issue1.htm (cit. on pp. 10, 18, 23, 28, 33, 34, 164, 166).

[69] Serina Chang et al. "Mobility Network Models of Covid-19 Explain Inequities and Inform Reopening." In: *Nature* 589.7840 (2021), pp. 82–87. DOI: 10.1038/s41586-020-2923-3 (cit. on p. 42).

[70] David Chaum. "Blind Signatures for Untraceable Payments." In: *Proceedings of the 2nd Annual International Cryptology Conference, CRYPTO 1982, Santa Barbara, California, USA, August 23-25, 1982*. Plenum Press, New York, 1982, pp. 199–203. DOI: 10.1007/978-1-4757-0602-4_18 (cit. on p. 50).

[71] Kenny Chee. *Bill Limiting Police Use of TraceTogether Data to Serious Crimes Passed*. The Straits Times. Accessed: 2024-02-12. 2021. URL: https://www.straitstimes.com/singapore/politics/bill-limiting-use-of-tracetogether-for-serious-crimes-passed-with-govt-assurances (cit. on pp. 3, 43).

[72] Bo-Rong Chen and Yih-Chun Hu. "Mitigating Denial-Of-Service Attacks on Digital Contact Tracing: Poster Abstract." In: *Proceedings of the 18th ACM Conference on Embedded Networked Sensor Systems, SenSys 2020, Online, Japan, November 16-19, 2020*. ACM, 2020, pp. 770–771. DOI: 10.1145/3384419.3430599 (cit. on p. 32).

[73] Hao Chen, Kim Laine, and Peter Rindal. "Fast Private Set Intersection from Homomorphic Encryption." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 1243–1255. DOI: 10.1145/3133956.3134061 (cit. on p. 46).

[74] Juliana Chen, Adrian Bauman, and Margaret Allman-Farinelli. "A Study to Determine the Most Popular Lifestyle Smartphone Applications and Willingness of the Public to Share Their Personal Data for Health Research." In: *Telemedicine and e-Health* 22.8 (2016), pp. 655–665. DOI: 10.1089/tmj.2015.0159 (cit. on p. 108).

[75] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. "Contact Tracing Mobile Apps for Covid-19: Privacy Considerations and Related Trade-Offs." In: *Computing Research Repository (CoRR)* abs/2003.11511 (2020). URL: https://arxiv.org/abs/2003.11511 (cit. on pp. 18, 25, 38, 45, 49, 62, 165, 166).

[76] Michele Ciampi and Claudio Orlandi. "Combining Private Set-Intersection with Secure Two-Party Computation." In: *Proceedings of the 11th International Conference Security and Cryptography for Networks, SCN 2018, Amalfi, Italy, September 5-7, 2018*. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 464–482. DOI: 10.1007/978-3-319-98113-0_25 (cit. on p. 73).

[77] Nicholas Confessore. *Cambridge Analytica and Facebook: The Scandal and the Fallout so Far*. New York Times. Accessed: 2024-01-26. 2018. URL: https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html (cit. on p. 120).

[78] Wanshu Cong. "From Pandemic Control to Data-Driven Governance: The Case of China's Health Code." In: *Frontiers in Political Science* 3 (2021), p. 627959. DOI: 10.3389/fpos.2021.627959 (cit. on pp. 42, 43, 91).

[79] Corona-Warn-App Project. *Availability*. Accessed: 2024-01-11. 2023. URL: https://www.coronawarn.app/en/faq/#availability (cit. on p. 23).

[80] Corona-Warn-App Project. *Kennzahlen zur Corona Warn App (Stand 10. Juni 2021)*. Accessed: 2024-01-12. 2021. URL: https://coronawarn.app/assets/documents/2021-06-10-cwa-daten-fakten.pdf (cit. on p. 86).

[81] Gil Correia, Lisa Rodrigues, Manuel Carlos Gameiro Da Silva, and Teresa M.F.O. Gonçalves. "Airborne Route and Bad Use of Ventilation Systems As Non-Negligible Factors in SARS-CoV-2

Transmission." In: *Medical Hypotheses* 141 (2020), p. 109781. DOI: 10.1016/j.mehy.2020.109781 (cit. on pp. 88, 94).

[82] Henry Corrigan-Gibbs and Dan Boneh. "Prio: Private, Robust, and Scalable Computation of Aggregate Statistics." In: *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*. USENIX Association, 2017, pp. 259–282. URL: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs (cit. on p. 134).

[83] Victor Costan and Srinivas Devadas. "Intel SGX Explained." In: *IACR Cryptology ePrint Archive* (2016), p. 86. URL: http://eprint.iacr.org/2016/086 (cit. on p. 127).

[84] Mick P Couper, Christopher Antoun, and Aigul Mavletova. "Total Survey Error in Practice." In: John Wiley & Sons Hoboken, NJ, 2017. Chap. 7 (cit. on p. 114).

[85] Cristina Criddle and Leo Kelion. *Coronavirus Contact-Tracing: World Split between Two Types of App*. BBC News. Accessed: 2024-01-11. 2020. URL: https://www.bbc.com/news/technology-52355028 (cit. on p. 20).

[86] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. "Fast and Private Computation of Cardinality of Set Intersection and Union." In: *Proceedings of the 11th International Conference on Cryptology and Network Security, CANS 2012, Darmstadt, Germany, December 12-14, 2012*. Vol. 7712. Springer, 2012, pp. 218–231. DOI: 10.1007/978-3-642-35404-5_17 (cit. on p. 66).

[87] David E. Culler et al. "Covista: A Unified View on Privacy Sensitive Mobile Contact Tracing Effort." In: *IEEE Data Engineering Bulletin* 45 (2 2020), pp. 83–94. URL: http://sites.computer.org/debull/A20june/issue1.htm (cit. on p. 92).

[88] George Danezis, Roger Dingledine, and Nick Mathewson. "Mixminion: Design of a Type III Anonymous Remailer Protocol." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2003, Berkeley, CA, USA, 11-14 May 2003*. IEEE Computer Society, 2003, pp. 2–15. DOI: 10.1109/SECPRI.2003.1199323 (cit. on p. 37).

[89] Paola Daniore, Tala Ballouz, Dominik Menges, and Viktor von Wyl. "The SwissCovid Digital Proximity Tracing App After One Year: Were Expectations Fulfilled?" In: *Swiss Medical Weekly* 151.3536 (2021), pp. 35–36. DOI: 10.4414/smw.2021.w30031 (cit. on pp. 40, 41).

[90] Helen Davidson. *China's Coronavirus Health Code Apps Raise Concerns Over Privacy*. The Guardian. Accessed: 2024-01-11. 2020. URL: https://www.theguardian.com/world/2020/apr/01/chinas-coronavirus-health-code-apps-raise-concerns-over-privacy (cit. on p. 39).

[91] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. "Unique in the Crowd: The Privacy Bounds of Human Mobility." In: *Scientific Reports* 3.1 (2013), pp. 1–5. DOI: 10.1038/srep01376 (cit. on p. 71).

[92] Didem Demirag and Erman Ayday. "Tracking and Controlling the Spread of a Virus in a Privacy-Preserving Way." In: *Computing Research Repository (CoRR)* abs/2003.13073 (2020). URL: https://arxiv.org/abs/2003.13073 (cit. on pp. 18, 20, 66, 163).

[93] Daniel Demmler, Thomas Schneider, and Michael Zohner. "ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation." In: *Proceedings of the 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*. The Internet Society, 2015. URL: https://www.ndss-symposium.org/ndss2015/aby---framework-efficient-mixed-protocol-secure-two-party-computation (cit. on p. 80).

[94] Der Spiegel. *Luca-App soll Personalausweise speichern und Bezahlen ermöglichen*. Der Spiegel. Accessed: 2024-02-15. 2022. URL: https://www.spiegel.de/netzwelt/apps/luca-app-soll-personalausweise-speichern-und-bezahlen-ermoeglichen-neues-geschaeftsmodell-a-ede036e6-0ed0-4148-9456-112fb1368fac (cit. on pp. 42, 90).

[95] Deutsche Telekom AG and SAP SE. *Corona-Warn-App*. Accessed: 2024-01-11. 2020. URL: https://github.com/corona-warn-app/cwa-documentation (cit. on pp. 23, 42).

[96] Deutsche Welle. *Coronavirus Tracking Apps: How Are Countries Monitoring Infections?* Accessed: 2024-01-11. 2020. URL: https://www.dw.com/en/coronavirus-tracking-apps-how-are-countries-monitoring-infections/a-53254234 (cit. on p. 9).

[97] Deutsche Welle. *Several German States Halt Covid Contact Tracing*. Accessed: 2024-02-14. 2021. URL: https://www.dw.com/en/germany-covid-contact-tracing-halted-in-several-states/a-60249871 (cit. on p. 41).

[98] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. *The Nym Network*. Accessed: 2024-01-12. 2021. URL: https://nymtech.net/nym-whitepaper.pdf (cit. on pp. 37, 53, 123, 125, 126).

[99] Whitfield Diffie and Martin Hellman. "New Directions in Cryptography." In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638 (cit. on pp. 47, 95).

[100] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. "Tor: The Second-Generation Onion Router." In: *Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA, August 9-13, 2004*. USENIX, 2004, pp. 303–320. URL: https://usenix.org/publications/library/proceedings/sec04/tech/full_papers/dingledine/dingledine.pdf (cit. on p. 37).

[101] Directorate of Health Iceland and Department of Civil Protection and Emergency Management Iceland. *Join the Tracing Team! Contagion Tracing Is a Community Affair*. Accessed: 2024-01-11. 2020. URL: https://web.archive.org/web/20201203170816/www.covid.is/app/en (cit. on p. 10).

[102] Distributed Ledger Technology Collaboration. *sse2neon*. Accessed: 2024-01-12. Dec. 2024. URL: https://github.com/DLTcollab/sse2neon (cit. on p. 80).

[103] DistriNet Research Group, KU Leuven. *LINDDUN Privacy Engineering*. Accessed: 2024-01-26. 2020. URL: https://www.linddun.org (cit. on p. 116).

[104] Jack Doerner and Abhi Shelat. "Scaling ORAM for Secure Computation." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 523–535. DOI: 10.1145/3133956.3133967 (cit. on p. 68).

[105] Brian Dolan. *SIG Introduces Bluetooth Low Energy Wireless Technology, the Next Generation of Bluetooth Wireless Technology*. Accessed: 2024-01-11. 2009. URL: https://www.mobihealthnews.com/5828/sig-introduces-bluetooth-low-energy-wireless-technology-the-next-generation-of-bluetooth-wireless-technology (cit. on p. 14).

[106] Marie Douriez, Harish Doraiswamy, Juliana Freire, and Cláudio T. Silva. "Anonymizing NYC Taxi Data: Does It Matter?" In: *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Montreal, QC, Canada, October 17-19, 2016*. IEEE, 2016, pp. 140–148. DOI: 10.1109/DSAA.2016.21 (cit. on p. 120).

[107] Thai Duong, Duong Hieu Phan, and Ni Trieu. "Catalic: Delegated PSI Cardinality with Applications to Contact Tracing." In: *Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2020, Daejeon, South Korea, December 7-11, 2020*. Vol. 12493. Lec-

ture Notes in Computer Science. Springer, 2020, pp. 870–899. DOI: 10.1007/978-3-030-64840-4_29 (cit. on p. 87).

[108] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy." In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407. DOI: 10.1561/0400000042 (cit. on pp. 120, 131, 132, 146).

[109] Ken TD Eames and Matt J Keeling. "Contact Tracing and Disease Control." In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1533 (2003), pp. 2565–2571. DOI: 10.1098/rspb.2003.2554 (cit. on p. 10).

[110] Stephan Ellmann, Markus Maryschok, Oliver Schöffski, and Martin Emmert. "The German Covid-19 Digital Contact Tracing App: A Socioeconomic Evaluation." In: *International Journal of Environmental Research and Public Health* 19.21 (2022), p. 14318. DOI: 10.3390/ijerph192114318 (cit. on pp. 40, 41).

[111] Ahmed Elmokashfi et al. "Nationwide Rollout Reveals Efficacy of Epidemic Control through Digital Contact Tracing." In: *Nature Communications* 12.1 (2021), p. 5918. DOI: 10.1038/s41467-021-26144-8 (cit. on pp. 15, 41).

[112] ENCRYPTO Group, TU Darmstadt. *OPPRF-PSI*. Accessed: 2024-01-12, Version Used: 2020-12-20. 2022. URL: https://github.com/encryptogroup/OPPRF-PSI (cit. on p. 80).

[113] Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, and Dan Boneh. "Express: Lowering the Cost of Metadata-Hiding Communication with Cryptographic Privacy." In: *Proceedings of the 30th USENIX Security Symposium, USENIX Security 2021, Online, August 11-13, 2021*. USENIX Association, 2021, pp. 1775–1792. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/eskandarian (cit. on p. 124).

[114] Saba Eskandarian and Matei Zaharia. "ObliDB: Oblivious Query Processing for Secure Databases." In: *Proceedings of the 46th International Conference on Very Large Data Bases, VLDB 2020, Online, August 31–September 4, 2020*. Vol. 13. 2. 2019, pp. 169–183. DOI: 10.14778/3364324.3364331 (cit. on p. 133).

[115] European Data Protection Board. *EDPB Letter Concerning the European Commission's Draft Guidance on Apps Supporting the Fight against the Covid-19 Pandemic*. Accessed: 2024-01-11. 2020. URL: https://edpb.europa.eu/our-work-tools/our-documents/letters/edpb-letter-concerning-european-commissions-draft-guidance_en (cit. on pp. 3, 9, 39).

[116] David Evans, Vladimir Kolesnikov, and Mike Rosulek. "A Pragmatic Introduction to Secure Multi-Party Computation." In: *Foundations and Trends in Privacy and Security* 2.2-3 (2018), pp. 70–246. ISSN: 2474-1558. DOI: 10.1561/3300000019 (cit. on pp. 64, 84).

[117] Ramsey Faragher and Robert Harle. "An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications." In: *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation, ION GNSS+ 2014, Tampa, Florida, September 2014*. Institute of Navigation (ION), 2014, pp. 201–210. URL: https://www.ion.org/publications/abstract.cfm?articleID=12411 (cit. on p. 14).

[118] Ramsey Faragher and Robert Harle. "Location Fingerprinting with Bluetooth Low Energy Beacons." In: *IEEE Journal on Selected Areas in Communications* 33.11 (2015), pp. 2418–2428. DOI: 10.1109/JSAC.2015.2430281 (cit. on p. 14).

[119] Emma Farge and John Revill. *'Test, Test, Test': WHO Chief's Coronavirus Message to World*. Reuters. Accessed: 2024-05-01. 20230. URL: https://www.reuters.com/article/us-healthcare-coronavirus-who-idUSKBN2132S4/ (cit. on p. 29).

[120] Federal Government of Germany. *Corona-Warn-App - Frequently Asked Questions*. Accessed: 2024-01-11. 2020. URL: https://www.bundesregierung.de/breg-de/themen/corona-warn-app/corona-warn-app-englisch/corona-warn-app-faq-1758636 (cit. on p. 31).

[121] Federal Ministry of Health Germany. *Verordnung über die Ausdehnung der Meldepflicht nach § 6 Absatz 1 Satz 1 Nummer 1 und § 7 Absatz 1 Satz 1 des Infektionsschutzgesetzes auf Infektionen mit dem erstmals im Dezember 2019 in Wuhan/Volksrepublik China aufgetretenen neuartigen Coronavirus ("2019-nCoV")*. Accessed: 2024-01-11. 2020. URL: https://www.bundesgesundheitsministerium.de/fileadmin/Dateien/3_Downloads/Gesetze_und_Verordnungen/GuV/C/Eilverordnung_Meldepflicht_Coronavirus.pdf (cit. on pp. 10, 12).

[122] Federal Trade Commission. *FTC Staff Report Finds Many Internet Service Providers Collect Troves of Personal Data, Users Have Few Options to Restrict Use*. Accessed: 2023-05-08. 2018. URL: https://www.ftc.gov/news-events/news/press-releases/2021/10/ftc-staff-report-finds-many-internet-service-providers-collect-troves-personal-data-users-have-few (cit. on p. 122).

[123] Susanne Felsen, Ágnes Kiss, Thomas Schneider, and Christian Weinert. "Secure and Private Function Evaluation with Intel SGX." In: *Proceedings of the ACM SIGSAC Conference on*

Cloud Computing Security Workshop, CCSW@CCS 2019, London, UK, November 11, 2019. ACM, 2019, pp. 165–181. DOI: 10.1145/3338466.3358919 (cit. on p. 134).

[124] Luca Ferretti et al. "Quantifying SARS-CoV-2 Transmission Suggests Epidemic Control with Digital Contact Tracing." In: *Science* 368.6491 (2020), pp. 1–7. DOI: 10.1126/science.abb6936 (cit. on pp. 9, 10, 39, 40, 42).

[125] Simon M Firestone, Robert M Christley, Michael P Ward, and Navneet K Dhand. "Adding the Spatial Dimension to the Social Network Analysis of an Epidemic: Investigation of the 2007 Outbreak of Equine Influenza in Australia." In: *Preventive Veterinary Medicine* 106.2 (2012), pp. 123–135. DOI: 10.1016/j.prevetmed.2012.01.020 (cit. on p. 11).

[126] Kiva A. Fisher et al. *Community and Close Contact Exposures Associated with Covid-19 among Symptomatic Adults ≥ 18 Years in 11 Outpatient Health Care Facilities*. Accessed: 16. November 2020. 2020. URL: https://www.cdc.gov/mmwr/volumes/69/wr/mm6936a5.htm (cit. on p. 42).

[127] FluPhone Study Team. *FluPhone Project: Understanding Spread of Infectious Disease and Behavioural Responses*. Accessed: 2024-01-11. 2011. URL: https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/fluphone2/ (cit. on pp. 10, 11, 17).

[128] Jeana Frost, Ivar E Vermeulen, and Nienke Beekers. "Anonymity Versus Privacy: Selective Information Sharing in Online Cancer Communities." In: *Journal of Medical Internet Research* 16.5 (2014). DOI: 10.2196/jmir.2684 (cit. on p. 123).

[129] Benjamin Fuller et al. "SoK: Cryptographically Protected Database Search." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 172–191. DOI: 10.1109/SP.2017.10 (cit. on p. 134).

[130] Patrizia Gabellini, Mauro D'Aloisio, Matteo Fabiani, and Valerio Placidi. "A Large Scale Trajectory Dataset for Shopper Behaviour Understanding." In: *Proceedings of the New Trends in Image Analysis and Processing International Workshops, ICIAP 2019 - Workshop, Trento, Italy, September 9-10, 2019*. Springer. 2019, pp. 285–295. DOI: 10.1007/978-3-030-30754-7_29 (cit. on p. 100).

[131] Matheus E. Garbelini, Chundong Wang, Sudipta Chattopadhyay, Sumei Sun, and Ernest Kurniawan. "SweynTooth: Unleashing Mayhem Over Bluetooth Low Energy." In: *Proceedings of the USENIX Annual Technical Conference, USENIX ATC 2020, Online, July 15-17, 2020*. USENIX Association, 2020, pp. 911–925. URL:

https://www.usenix.org/conference/atc20/presentation/garbelini (cit. on p. 16).

[132] GDPR.EU. *What Are the GDPR Consent Requirements?* Accessed: 2024-01-26. 2024. URL: https://gdpr.eu/gdpr-consent-requirements (cit. on pp. 108, 113, 114, 120, 145).

[133] Hadi Givehchian et al. "Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices." In: *Proceedings of the 43rd IEEE Symposium on Security and Privacy, S&P 2022, San Francisco, CA, USA, May 22-26, 2022*. IEEE, 2022, pp. 1690–1704. DOI: 10.1109/SP46214.2022.9833758 (cit. on pp. 16, 35).

[134] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play Any Mental Game or A Completeness Theorem for Protocols with Honest Majority." In: *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, SIGACT 1987, New York, New York, USA, 1987*. ACM, 1987, pp. 218–229. DOI: 10.1145/28395.28420 (cit. on pp. 21, 70).

[135] Google. *Analyze Power Use with Battery Historian*. Accessed: 2024-01-12. 2024. URL: https://developer.android.com/topic/performance/power/battery-historian (cit. on p. 82).

[136] Google. *Exposure Notifications API Service Update*. Accessed: 2024-01-12. 2024. URL: https://developers.google.com/android/exposure-notifications (cit. on p. 23).

[137] Google. *Exposure Notifications Verification Server*. Accessed: 2024-01-11. 2020. URL: https://web.archive.org/web/20201126122752/developers.google.com/android/exposure-notifications/verification-system (cit. on p. 27).

[138] Google. *Manage Your Location History*. Accessed: 2024-01-30. 2024. URL: https://support.google.com/accounts/answer/3118687?hl=en (cit. on p. 69).

[139] Google and Apple. *Eposure Notification Privacy-preserving Analytics (ENPA) White Paper*. Accessed: 2024-01-14. 2021. URL: https://github.com/google/exposure-notifications-android/blob/022032546621496636 8b1f1e5cc0eb8a6a380df7/doc/ENPA.pdf (cit. on p. 30).

[140] Google and Apple. *Exposure Notification - Bluetooth Specification*. Accessed: 2024-01-11. 2020. URL: https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf (cit. on pp. 27, 34).

[141] Google and Apple. *Privacy-Preserving Contact Tracing*. Accessed: 2024-01-11. 2020. URL: https://covid19.apple.com/contacttracing (cit. on pp. 18, 22, 23, 28, 32, 34, 164, 166).

[142] Government of Singapore. *BlueTrace Protocol - Privacy-Preserving Cross-Border Contact Tracing*. Accessed: 2024-01-11. 2020. URL: https://bluetrace.io (cit. on pp. 10, 15, 18–20, 26, 31–34, 38, 39, 163, 166).

[143] Ashish Gupta and Anil Dhami. "Measuring the Impact of Security, Trust and Privacy in Information Sharing: A Study on Social Networking Sites." In: *Journal of Direct, Data and Digital Marketing Practice* 17 (2015), pp. 43–53. DOI: 10.1057/dddmp.2015.32 (cit. on p. 127).

[144] Yaron Gvili. "Security Analysis of the Covid-19 Contact Tracing Specifications by Apple Inc. and Google Inc." In: *IACR Cryptology ePrint Archive* (2020), p. 428. URL: https://eprint.iacr.org/2020/428 (cit. on pp. 32–34, 36).

[145] Marieke Haan, Peter Lugtig, and Vera Toepoel. "Can We Predict Device Use? An Investigation into Mobile Device Use in Surveys." In: *International Journal of Social Research Methodology* 22.5 (2019), pp. 517–531. DOI: 10.1080/13645579.2019.1593340 (cit. on p. 114).

[146] Nick Hajli and Xiaolin Lin. "Exploring the Security of Information Sharing on Social Networking Sites: The Role of Perceived Control of Information." In: *Journal of Business Ethics* 133 (2016), pp. 111–123. DOI: 10.1007/s10551-014-2346-x (cit. on p. 127).

[147] David M. Halbfinger, Isabel Kershner, and Ronen Bergman. *To Track Coronavirus, Israel Moves to Tap Secret Trove of Cellphone Data*. New York Times. Accessed: 2024-01-23. 2020. URL: https://www.nytimes.com/2020/03/16/world/middleeast/israel-coronavirus-cellphone-tracking.html?referringSource=articleShare (cit. on pp. 3, 9, 16, 43, 67).

[148] Shima Hamidi and Ahoura Zandiatashbar. "Compact Development and Adherence to Stay-At-Home Order during the Covid-19 Pandemic: A Longitudinal Investigation in the United States." In: *Landscape and Urban Planning* 205 (2021), p. 103952. DOI: 10.1016/j.landurbplan.2020.103952 (cit. on p. 127).

[149] Isobel Asher Hamilton. *Compulsory Selfies and Contact-Tracing: Authorities Everywhere Are Using Smartphones to Track the Coronavirus, and It's Part of a Massive Increase in Global Surveillance*. Business Insider. Accessed: 2024-01-11. 2020. URL: https://www.businessinsider.com/countries-tracking-citizens-phones-coronavirus-2020-3?r=DE%5C&IR=T (cit. on pp. 3, 9, 42, 43).

[150] Lea Hamner et al. *High SARS-CoV-2 Attack Rate Following Exposure at a Choir Practice*. Accessed: 2024-03-12. 2020. URL: https://www.cdc.gov/mmwr/volumes/69/wr/mm6919e6.htm (cit. on p. 42).

[151] Marc Hasselwander et al. "Building Back Better: The Covid-19 Pandemic and Transport Policy Implications for a Developing Megacity." In: *Sustainable Cities and Society* 69 (2021), p. 102864. DOI: 10.1016/j.scs.2021.102864 (cit. on p. 127).

[152] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. "Who Can Find My Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System." In: *Proceedings on Privacy Enhancing Technologies, PETs 2021, Online, July 12–16, 2021*. 2021. DOI: 10.2478/POPETS-2021-0045 (cit. on pp. 51, 52).

[153] Arvin Hekmati, Gowri Sankar Ramachandran, and Bhaskar Krishnamachari. "CONTAIN: Privacy-Oriented Contact Tracing Protocols for Epidemics." In: *Proceedings of the 17th IFIP/IEEE International Symposium on Integrated Network Management, IM 2021, Bordeaux, France, May 17-21, 2021*. IEEE, 2021, pp. 872–877. URL: https://ieeexplore.ieee.org/document/9464051 (cit. on pp. 18, 23, 28, 38, 164, 166).

[154] Stephen Hemminger, Fabio Ludovici, and Hagen Paul Pfeifer. *Tc-Netem(8) - Linux Manual Page*. Accessed: 2024-01-12. 2011. URL: https://man7.org/linux/man-pages/man8/tc-netem.8.html (cit. on p. 80).

[155] Robert Hinch et al. *Effective Configurations of a Digital Contact Tracing App: A Report to NHSX*. Accessed: 2024-01-11. 2020. URL: https://github.com/BDI-pathogens/covid-19_instant_tracing (cit. on p. 38).

[156] Peter J Hotez et al. "America's deadly flirtation with antiscience and the medical freedom movement." In: *The Journal of Clinical Investigation* 131.7 (2021). DOI: 10.1172/JCI149072 (cit. on p. 126).

[157] Patrick Howell O'Neill. *No, Coronavirus Apps Don't Need 60% Adoption to Be Effective*. MIT Technology Review. Accessed: 2024-01-11. 2020. URL: https://www.technologyreview.com/2020/06/05/1002775/covid-apps-effective-at-less-than-60-percent-download/ (cit. on pp. 3, 39).

[158] Yan Huang, David Evans, and Jonathan Katz. "Private Set Intersection: Are Garbled Circuits Better than Custom Protocols?" In: *Proceedings of the 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012. URL: https://www.ndss-symposium.org/ndss2012/private-set-intersection-are-garbled-circuits-better-custom-protocols (cit. on p. 84).

[159] Zhilian Huang et al. "Performance of Digital Contact Tracing Tools for Covid-19 Response in Singapore: Cross-Sectional Study." In: *JMIR mHealth and uHealth* 8.10 (2020). DOI: 10.2196/23148 (cit. on p. 17).

[160] Ramon Huerta and Lev S Tsimring. "Contact Tracing and Epidemics Control in Social Networks." In: *Physical Review E* 66.5 (2002), pp. 1–4. DOI: 10.1103/PhysRevE.66.056115 (cit. on p. 10).

[161] Inria. *ROBust and Privacy-PresERving Proximity Tracing Protocol*. Accessed: 2024-01-11. 2020. URL: https://github.com/ROBERT-proximity-tracing/documents (cit. on pp. 18, 27, 30, 33, 36–38, 163, 166).

[162] Intel. *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*. Tech. rep. Accessed: 2024-01-29. 2023. URL: https://cdrdv2.intel.com/v1/dl/getContent/733568 (cit. on p. 130).

[163] Intel. *From ARM NEON* to Intel SSE*. Accessed: 2024-01-12. 2021. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/arm-neon-to-sse-automatic-porting-solution-tips-and-tricks.html (cit. on p. 80).

[164] Intel. *Intel® TDX*. Accessed: 2024-01-26. 2024. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-trust-domain-extensions.html (cit. on pp. 127, 130).

[165] Intel. *Intel® TDX Module Base Architecture Specification*. Tech. rep. Accessed: 2024-01-26. Intel, 2023. URL: https://cdrdv2.intel.com/v1/dl/getContent/733575 (cit. on pp. 129, 130).

[166] Intel. *MKTME Side Channel Impact on Intel TDX*. Accessed: 2024-01-29. 2024. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/best-practices/mktme-side-channel-impact-on-intel-tdx.html (cit. on pp. 130, 137).

[167] iPerf Project. *iPerf - The TCP, UDP and SCTP Network Bandwidth Measurement Tool*. Accessed: 2024-01-12. 2024. URL: https://iperf.fr (cit. on p. 81).

[168] Leonardo Horn Iwaya, M Ali Babar, Awais Rashid, and Chamila Wijayarathna. "On the Privacy of Mental Health Apps: An Empirical Investigation and Its Implications for App Development." In: *Empirical Software Engineering* 28.1 (2023), p. 2. DOI: 10.1007/s10664-022-10236-0 (cit. on p. 109).

[169] Seungyeon Jeong, Seungho Kuk, and Hyogon Kim. "A Smartphone Magnetometer-Based Diagnostic Test for Automatic Contact Tracing in Infectious Disease Epidemics." In: *IEEE Access* 7 (2019), pp. 20734–20747. DOI: 10.1109/ACCESS.2019.2895075 (cit. on p. 16).

[170] Mahabir Prasad Jhanwar and Sumanta Sarkar. "PHyCT: Privacy Preserving Hybrid Contact Tracing." In: *IACR Cryptology ePrint Archive* (2020), p. 793. URL: https://eprint.iacr.org/2020/793 (cit. on pp. 18, 19, 33, 163, 166).

[171] Johns Hopkins University. *The Johns Hopkins Foreign Affairs Symposium Presents: The Price of Privacy: Re-Evaluating the NSA*. Youtube. Accessed: 2024-02-23, Quote at Timestamp: 17:59 min. 2014. URL: https://www.youtube.com/watch?v=kV2HDM86XgI (cit. on p. 6).

[172] Yann Joly, Gratien Dalpé, Derek So, and Stanislav Birko. "Fair Shares and Sharing Fairly: A Survey of Public Views on Open Science, Informed Consent and Participatory Research in Biobanking." In: *PLOS ONE* 10.7 (July 2015), pp. 1–20. DOI: 10.1371/journal.pone.0129893 (cit. on p. 114).

[173] Otso Jousimaa. *Bluetooth Beacon Density Maximum*. Accessed: 2024-01-11. 2020. URL: https://ruuvi.com/bluetooth-beacon-maximum-density/ (cit. on p. 32).

[174] Gyuwon Jung, Hyunsoo Lee, Auk Kim, and Uichin Lee. "Too Much Information: Assessing Privacy Risks of Contact Trace Data Disclosure on People with Covid-19 in South Korea." In: *Frontiers in Public Health* 8 (2020), p. 305. DOI: 10.3389/fpubh.2020.00305 (cit. on p. 71).

[175] Kaggle. *kaggle*. Accessed: 2024-01-26. June 2023. URL: https://www.kaggle.com (cit. on p. 153).

[176] Seny Kamara, Payman Mohassel, and Mariana Raykova. "Outsourcing Multi-Party Computation." In: *IACR Cryptology ePrint Archive* (2011), p. 272. URL: http://eprint.iacr.org/2011/272 (cit. on p. 30).

[177] Seny Kamara and Mariana Raykova. "Secure Outsourced Computation in a Multi-Tenant Cloud." In: *Proceedings of the Workshop on Cryptography and Security in Clouds 2011, Zurich, Switzerland, March 15-16, 2011*. 2011, pp. 1–5. URL: https://cachin.com/cc/csc2011/submissions/kamara.pdf (cit. on p. 134).

[178] Seny Kamara et al. "SoK: Cryptanalysis of Encrypted Search with LEAKER - A Framework for Leakage Attack Evaluation on Real-World Data." In: *Proceedings of the 7th IEEE European Symposium on Security and Privacy, EuroS&P 2022, Genoa, Italy, June 6-10, 2022*. IEEE, 2022, pp. 90–108. DOI: 10.1109/EUROSP53844.2022.00014 (cit. on pp. 127, 134).

[179] Malek Karaim, Mohamed Elsheikh, and Aboelmagd Noureldin. "GNSS Error Sources." In: London, UK: IntechOpen, May 2018. Chap. 2. ISBN: 978-1-78923-215-8. DOI: 10.5772/intechopen.75493 (cit. on p. 16).

[180] Matt J Keeling, T Deirdre Hollingsworth, and Jonathan M Read. "Efficacy of Contact Tracing for the Containment of the 2019 Novel Coronavirus (Covid-19)." In: *Journal of Epidemiology and Community Health* 74.10 (2020), pp. 861–866. DOI: 10.1136/jech-2020-214051 (cit. on p. 86).

[181] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. "Generic Attacks on Secure Outsourced Databases." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2016, Vienna, Austria, October 24-28, 2016*. ACM, 2016, pp. 1329–1340. DOI: 10.1145/2976749.2978386 (cit. on pp. 127, 131, 134, 137, 171).

[182] Michelle Kendall et al. "Epidemiological Impacts of the NHS Covid-19 App in England and Wales Throughout Its First Year." In: *Nature Communications* 14.1 (2023), p. 858. DOI: 10.1038/s41467-023-36495-z (cit. on pp. 40, 41).

[183] Christopher T. Kenny et al. "The Use of Differential Privacy for Census Data and Its Impact on Redistricting: The Case of the 2020 U.S. Census." In: *Science Advances* 7.41 (2021). DOI: 10.1126/sciadv.abk3283 (cit. on p. 120).

[184] Florian Keusch, Bella Struminskaya, Christopher Antoun, Mick P Couper, and Frauke Kreuter. "Willingness to Participate in Passive Mobile Data Collection." In: *Public Opinion Quarterly* 83.S1 (2019), pp. 210–235. DOI: 10.1093/poq/nfz007 (cit. on pp. 113, 114).

[185] Ágnes Kiss, Jian Liu, Thomas Schneider, N. Asokan, and Benny Pinkas. "Private Set Intersection for Unequal Set Sizes with Mobile Applications." In: *Proceedings on Privacy Enhancing Technologies, PETS 2017, Minneapolis, USA, July 18 – 21, 2017*. 2017. DOI: 10.1515/POPETS-2017-0044 (cit. on p. 21).

[186] Michael Klenk and Hein Duijf. "Ethics of Digital Contact Tracing and Covid-19: Who Is (not) Free to Go?" In: *Ethics and Information Technology* 23.S1 (2021), pp. 69–77. DOI: 10.1007/s10676-020-09544-0 (cit. on pp. 9, 43).

[187] Patrick Koeberl et al. "Time to Rethink: Trust Brokerage Using Trusted Execution Environments." In: *Proceedings of the 8th International Conference on Trust and Trustworthy Computing, TRUST 2015, Heraklion, Greece, August 24-26, 2015*. Vol. 9229. Lecture Notes in Computer Science. Springer, 2015, pp. 181–190. DOI: 10.1007/978-3-319-22846-4_11 (cit. on p. 133).

[188] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. "Efficient Batched Oblivious PRF with Applications to Private Set Intersection." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2016, Vienna, Austria, October 24-28, 2016*. ACM, 2016, pp. 818–829. DOI: 10.1145/2976749.2978381 (cit. on p. 80).

[189] Marvin Kowalewski et al. "52 Weeks Later: Attitudes Towards Covid-19 Apps for Different Purposes Over Time." In: *Proceedings of the ACM on Human-Computer Interaction* 7.CSCW2 (2023), pp. 1–45. DOI: 10.1145/3610042 (cit. on pp. 3, 31, 39, 40, 42, 43).

[190] Simeon Krastnikov, Florian Kerschbaum, and Douglas Stebila. "Efficient Oblivious Database Joins." In: *Proceedings of the 46th International Conference on Very Large Data Bases, VLDB 2020, Online, August 31–September 4, 2020*. Vol. 13. 11. 2020, pp. 2132–2145. DOI: 10.14778/3407790.3407814 (cit. on pp. 138, 147).

[191] Mirjam E Kretzschmar et al. "Impact of Delays on Effectiveness of Contact Tracing Strategies for Covid-19: A Modelling Study." In: *The Lancet Public Health* 5.8 (2020). DOI: 10.1016/S2468-2667(20)30157-2 (cit. on pp. 39, 41).

[192] Adam J Kucharski et al. "Effectiveness of Isolation, Testing, Contact Tracing, and Physical Distancing on Reducing Transmission of SARS-CoV-2 in Different Settings: A Mathematical Modelling Study." In: *The Lancet Infectious Diseases* 20.10 (2020), pp. 1151–1160. DOI: 10.1016/S1473-3099(20)30457-6 (cit. on p. 38).

[193] Seungho Kuk, Junha Kim, Yongtae Park, and Hyogon Kim. "Empirical Determination of Efficient Sensing Frequencies for Magnetometer-Based Continuous Human Contact Monitoring." In: *Sensors* 18.5 (2018), p. 1358. DOI: 10.3390/s18051358 (cit. on p. 11).

[194] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. "Improved Reconstruction Attacks on Encrypted Data Using Range Query Leakage." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2018, Proceedings, San Francisco, California, USA,21-23 May 2018*. IEEE Computer Society, 2018, pp. 297–314. DOI: 10.1109/SP.2018.00002 (cit. on pp. 127, 131, 134, 137, 171).

[195] Christine Lambrecht and Silvia Engels. *Bundesjustizministerin: Handy-Tracking geht "nur mit Freiwilligkeit"*. Deutschlandfunk. Accessed: 2024-01-11. 2020. URL: https://www.deutschlandfunk.de/corona-pandemie-bundesjustizministerin-handy-tracking-geht-100.html (cit. on pp. 3, 9, 39).

[196] Susan Landau. "Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations." In: *IEEE Security & Privacy* 11.4 (2013), pp. 54–63. DOI: 10.1109/MSP.2013.90 (cit. on p. 123).

[197] Thomas Lecocq et al. "Global Quieting of High-Frequency Seismic Noise Due to Covid-19 Pandemic Lockdown Measures." In: *Science* 369.6509 (2020), pp. 1338–1343. DOI: 10.1126/science.abd2438 (cit. on p. 127).

[198] Douglas J Leith and Stephen Farrell. "Measurement-Based Evaluation of Google/Apple Exposure Notification API for Proximity Detection in a Light-Rail Tram." In: *PLOS ONE* 15.9 (2020). DOI: 10.1371/journal.pone.0239943 (cit. on p. 15).

[199] Xavier Leroy. *CompCert*. Accessed: 2024-01-29. 2023. URL: https://compcert.org/index.html (cit. on p. 137).

[200] Dyani Lewis. *Why Many Countries Failed at Covid-19 Contact-Tracing–but Some Got It Right*. Nature News Feature. Accessed: 2024-02-15. 2020. URL: https://www.nature.com/articles/d41586-020-03518-4 (cit. on p. 41).

[201] Veronica QT Li, Liang Ma, and Xun Wu. "Covid-19, Policy Change, and Post-Pandemic Data Governance: A Case Analysis of Contact Tracing Applications in East Asia." In: *Policy and Society* 41.1 (2022), pp. 129–142. DOI: 10.1093/polsoc/puab019 (cit. on pp. 42, 43).

[202] Alessandro Liberati et al. "The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration." In: *Annals of Internal Medicine* 151.4 (2009). DOI: 10.1136/bmj.b2700 (cit. on p. 111).

[203] Matthias Linden et al. "Case Numbers beyond Contact Tracing Capacity Are Endangering the Containment of Covid-19." In: *Deutsches Ärzteblatt International* 117.46 (2020), p. 790. DOI: 10.3238/arztebl.2020.0790 (cit. on p. 40).

[204] Jonathan Lis. *About 60 Percent of Israelis' Appeals against Quarantine Based on Digital Tracking Granted*. Haaretz. Accessed: 2024-02-13. 2020. URL: https://www.haaretz.com/israel-news/2020-07-20/ty-article/.premium/about-60-percent-of-appeals-against-quarantine-based-on-digital-tracking-granted/0000017f-e0e3-d9aa-afff-f9fbb0a80000 (cit. on p. 72).

[205] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yan Huang, and Elaine Shi. "ObliVM: A Programming Framework for Secure Computation." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2015, San Jose, CA, USA, May 17-21, 2015*. IEEE

Computer Society, 2015, pp. 359–376. DOI: 10.1109/SP.2015.29 (cit. on p. 134).

[206] Chang Liu et al. "Ghostrider: A Hardware-Software System for Memory Trace Oblivious Computation." In: *Proceedings of the 20th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2015, Istanbul, Turkey, March 14-18, 2015*. ACM, 2015, pp. 87–101. DOI: 10.1145/2694344.2694385 (cit. on p. 134).

[207] Jia Liu, Canfeng Chen, and Yan Ma. "Modeling Neighbor Discovery in Bluetooth Low Energy Networks." In: *IEEE Communication Letters* 16.9 (2012), pp. 1439–1441. DOI: 10.1109/LCOMM.2012.073112.120877 (cit. on p. 14).

[208] Joseph K. Liu and Duncan S. Wong. "On the Security Models of (Threshold) Ring Signature Schemes." In: *Proceedings of the 7th International Conference Information Security and Cryptology, ICISC 2004, Seoul, Korea, December 2-3, 2004*. Vol. 3506. Lecture Notes in Computer Science. Springer, 2004, pp. 204–217. DOI: 10.1007/11496618_16 (cit. on p. 57).

[209] Joseph K. Liu et al. "Privacy-Preserving Covid-19 Contact Tracing App: A Zero-Knowledge Proof Approach." In: *IACR Cryptology ePrint Archive* (2020), p. 528. URL: https://eprint.iacr.org/2020/528 (cit. on p. 29).

[210] Shu Liu, Yingxin Jiang, and Aaron Striegel. "Face-To-Face Proximity Estimation Using Bluetooth on Smartphones." In: *IEEE Transactions on Mobile Computing* 13.4 (2013), pp. 811–823. DOI: 10.1109/TMC.2013.44 (cit. on p. 13).

[211] Xuanzhe Liu et al. "Understanding Diverse Usage Patterns from Large-Scale Appstore-Service Profiles." In: *IEEE Transactions on Software Engineering* 44.4 (2018), pp. 384–411. DOI: 10.1109/TSE.2017.2685387 (cit. on p. 122).

[212] Lucien Loiseau et al. *Whisper Tracing Version 3 - an Open and Privacy First Protocol for Contact Tracing*. Accessed: 2024-01-11. 2020. URL: https://docsend.com/view/nis3dac (cit. on pp. 18, 24, 25, 28, 48, 164–166).

[213] Wouter Lueks et al. "CrowdNotifier: Decentralized Privacy-Preserving Presence Tracing." In: *Proceedings on Privacy Enhancing Technologies, PETS 2021, Online, July 12–16, 2021*. 2021. DOI: 10.2478/POPETS-2021-0074 (cit. on pp. 42, 91).

[214] Kim Lyons. *Clear Channel's Billboards Will Start Tracking Consumers in Europe*. The Verge. Accessed: 2024-01-16. 2020. URL: https://www.theverge.com/2020/8/10/21361734/clear-channel-billboards-privacy-ad-tracking-europe (cit. on p. 34).

[215] Jackie Ma et al. *Proximity Tracing App: Report from the Measurement Campaing 2020-04-09*. Accessed: 2024-01-11. 2020. URL: https://github.com/pepp-pt/pepp-pt-documentation/blob/master/12-proximity-measurement/2020-04-09-BW-report-epi-mod.pdf (cit. on p. 15).

[216] Macworld. *iPhone Vs Android Market Share*. Accessed: 2024-01-11. 2019. URL: https://www.macworld.com/article/673487/iphone-vs-android-market-share.html (cit. on p. 31).

[217] Parag Mahale et al. *Multiple Covid-19 Outbreaks Linked to a Wedding Reception in Rural Maine*. Accessed: 2024-03-12. 2020. URL: https://www.cdc.gov/mmwr/volumes/69/wr/mm6945a5.htm (cit. on p. 42).

[218] Marie-Helen Maras, Michelle D Miranda, and Adam Scott Wandt. "The Use of Covid-19 Contact Tracing App Data As Evidence of a Crime." In: *Science & Justice* 63.2 (2023), pp. 158–163. DOI: 10.1016/j.scijus.2022.12.008 (cit. on pp. 3, 43).

[219] MariaDB Foundation. *MariaDB Server: The Open Source Relational Database*. Accessed: 2023-12-05. 2023. URL: https://mariadb.org/ (cit. on p. 150).

[220] Massachusetts Institute of Technology. *Project Safe Paths*. Accessed: 2024-01-11. 2020. URL: https://www.media.mit.edu/projects/safepaths/overview/ (cit. on p. 16).

[221] Petar Maymounkov and David Mazières. "Kademlia: A Peer-To-Peer Information System Based on the XOR Metric." In: *Proceedings of the First International Workshop on Peer-to-Peer Systems, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002*. Vol. 2429. Lecture Notes in Computer Science. Springer, 2002, pp. 53–65. DOI: 10.1007/3-540-45748-8_5 (cit. on pp. 53, 54).

[222] Ministry of Health New Zealand. *NZ COVID Tracer App Evolving*. Accessed: 2024-02-06. 2022. URL: https://www.health.govt.nz/news-media/news-items/nz-covid-tracer-app-evolving (cit. on pp. 42, 90).

[223] Ministry of Health Singapore. *Two Charged under Infectious Diseases Act for False Information and Obstruction of Contact Tracing*. Accessed: 2024-01-11. 2020. URL: https://www.moh.gov.sg/news-highlights/details/two-charged-under-infectious-diseases-act-for-false-information-and-obstruction-of-contact-tracing (cit. on pp. 10, 12).

[224] Pratyush Mishra, Rishabh Poddar, Jerry Chen, Alessandro Chiesa, and Raluca Ada Popa. "Oblix: An Efficient Oblivious Search Index." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2018, San Francisco, California, USA, 21-23 May 2018*. IEEE Computer Society, 2018, pp. 279–296. DOI:

10.1109/SP.2018.00045 (cit. on pp. 127, 128, 133, 135, 136, 138, 139, 147, 149–151, 156, 171).

[225] John C. Mitchell and Joe Zimmerman. "Data-Oblivious Data Structures." In: *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science, STACS 2014, Lyon, France, March 5-8, 2014*. Vol. 25. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014, pp. 554–565. DOI: 10.4230/LIPICS.STACS.2014.554 (cit. on p. 131).

[226] MITRE Corporation. *Bluetooth : Security Vulnerabilities, CVEs*. Accessed: 2024-05-01. 2024. URL: https://www.cvedetails.com/vulnerability-list/vendor_id-11436/Bluetooth.html (cit. on p. 13).

[227] MITRE Corporation. *CVE-2019-2102*. Accessed: 2024-01-11. 2019. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-2102 (cit. on p. 16).

[228] Kenji Mizumoto, Katsushi Kagaya, Alexander Zarebski, and Gerardo Chowell. "Estimating the Asymptomatic Proportion of Coronavirus Disease 2019 (Covid-19) Cases on Board the Diamond Princess Cruise Ship, Yokohama, Japan, 2020." In: *Eurosurveillance* 25.10 (2020), p. 2000180. DOI: 10.2807/1560-7917.ES.2020.25.10.2000180 (cit. on p. 40).

[229] MobileCoin. *MobileCoin White Paper*. Accessed: 2024-01-14. 2023. URL: https://mobilecoin.com/learn/read-the-whitepapers/mobilecoin/ (cit. on p. 54).

[230] Alessandro Montanari. "Multimodal Indoor Social Interaction Sensing and Real-Time Feedback for Behavioural Intervention." In: *Proceedings of the Workshop on Wireless of the Students, by the Students, & for the Students, S3@MobiCom 2015, Paris, France, September 11, 2015*. ACM, 2015, pp. 7–9. DOI: 10.1145/2801694.2801706 (cit. on p. 14).

[231] Peter Murray-Rust. "Open Data in Science." In: *Nature Precedings* (2008), p. 1. DOI: 10.1038/npre.2008.1526.1 (cit. on p. 114).

[232] National Informatics Centre, Ministry of Electronics & Information Technology India. *Aarogya Setu Mobile App*. Accessed: 2024-01-11. 2020. URL: https://www.mygov.in/aarogya-setu-app/ (cit. on pp. 18, 19, 91, 163).

[233] National Library of Medicine. *PubMed*. Accessed: 2023-03-03. 2023. URL: https://pubmed.ncbi.nlm.nih.gov/ (cit. on pp. 109, 110).

[234] Muhammad Naveed. "The Fallacy of Composition of Oblivious RAM and Searchable Encryption." In: *IACR Cryptology ePrint Archive* (2015), p. 668. URL: http://eprint.iacr.org/2015/668 (cit. on p. 139).

[235] New York Times. *Lockdowns in France and U.K. Expected to Last into Next Month*. New York Times. Accessed: 2024-01-11. 2020. URL: https://www.nytimes.com/2020/04/13/world/coronavirus-news-world-international-global.html (cit. on p. 9).

[236] New York Times. *White House Is Not Tracing Contacts for 'Super-Spreader' Rose Garden Event*. Accessed: 2024-03-12. 2020. URL: https://www.nytimes.com/2020/10/05/health/contact-tracing-white-house.html (cit. on p. 42).

[237] Casey Newton. *Why Countries Keep Bowing to Apple and Google's Contact Tracing App Requirements*. The Verge. Accessed: 2024-01-12. 2020. URL: https://www.theverge.com/interface/2020/5/8/21250744/apple-google-contact-tracing-england-germany-exposure-notification-india-privacy (cit. on pp. 23, 31).

[238] Khuong An Nguyen, Chris Watkins, and Zhiyuan Luo. "Co-Location Epidemic Tracking on London Public Transports Using Low Power Mobile Magnetometer." In: *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017, Sapporo, Japan, September 18-21, 2017*. IEEE, 2017, pp. 1–8. DOI: 10.1109/IPIN.2017.8115963 (cit. on pp. 11, 16).

[239] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. "A Survey of Published Attacks on Intel SGX." In: *Computing Research Repository (CoRR)* abs/2006.13598 (2020). URL: https://arxiv.org/abs/2006.13598 (cit. on pp. 127, 129, 133).

[240] Rishab Nithyanand, Oleksii Starov, Phillipa Gill, Adva Zair, and Michael Schapira. "Measuring and Mitigating AS-Level Adversaries against Tor." In: *Proceedings of the 23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016. URL: https://www.ndss-symposium.org/wp-content/uploads/2017/09/measuring-mitigating-as-level-adversaries-against-tor.pdf (cit. on p. 37).

[241] Meir Nizri. *Covid-19 Dataset*. Accessed: 2024-01-29. 2023. URL: https://www.kaggle.com/datasets/meirnizri/covid19-dataset (cit. on p. 153).

[242] Leysan Nurgalieva, David O'Callaghan, and Gavin Doherty. "Security and Privacy of mHealth Applications: A Scoping Review." In: *IEEE Access* 8 (2020), pp. 104247–104268. DOI: 10.1109/ACCESS.2020.2999934 (cit. on p. 109).

[243] Andrea Nuzzo et al. "Universal Shelter-In-Place Versus Advanced Automated Contact Tracing and Targeted Isolation: A Case for 21st-Century Technologies for SARS-CoV-2 and Future

Pandemics." In: 95.9 (2020), pp. 1898–1905. DOI: 10.1016/j.mayocp.2020.06.027 (cit. on p. 38).

[244] Huthaifa A. Obeidat, Wafa Shuaieb, Omar Obeidat, and Raed Abd-Alhameed. "A Review of Indoor Localization Techniques and Wireless Technologies." In: *Wireless Personal Communications* 119.1 (2021), pp. 289–327. DOI: 10.1007/S11277-021-08209-5 (cit. on p. 16).

[245] Rasmus Pagh and Flemming Friche Rodler. "Cuckoo Hashing." In: *Proceedings of the 9th Annual European Symposium on Algorithms, ESA 2001, Aarhus, Denmark, August 28-31, 2001*. Vol. 2161. Lecture Notes in Computer Science. Springer, 2001, pp. 121–133. DOI: 10.1007/3-540-44676-1_10 (cit. on p. 74).

[246] Sarvar Patel, Giuseppe Persiano, Kevin Yeo, and Moti Yung. "Mitigating Leakage in Secure Cloud-Hosted Data Structures: Volume-Hiding for Multi-Maps via Hashing." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM, 2019, pp. 79–93. DOI: 10.1145/3319535.3354213 (cit. on p. 133).

[247] PePP-PT e.V. i.Gr. *PePP-PT Documentation*. Accessed: 2024-01-11. 2020. URL: https://github.com/pepp-pt/pepp-pt-documentation (cit. on pp. 18–20, 26, 30, 33, 37, 38, 163, 166).

[248] Krzysztof Pietrzak. "Delayed Authentication: Preventing Replay and Relay Attacks in Private Contact Tracing." In: *Proceedings of the 21st International Conference on Cryptology in India, INDOCRYPT 2020, Bangalore, India, December 13-16, 2020*. Vol. 12578. Lecture Notes in Computer Science. Springer, 2020, pp. 3–15. DOI: 10.1007/978-3-030-65277-7_1 (cit. on p. 33).

[249] Benny Pinkas and Eyal Ronen. "Hashomer – Privacy-Preserving Bluetooth Based Contact Tracing Scheme for Hamagen." In: *Proceedings of the NDSS Workshop on Secure IT Technologies against Covid-19, CoronaDef 2021, Online, February 21, 2021*. Internet Society, 2021, pp. 1–7. URL: https://www.ndss-symposium.org/ndss-program/coronadef-2021/ (cit. on pp. 18, 23, 28–31, 33, 36, 38, 46, 164, 166).

[250] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. "PSI from PaXoS: Fast, Malicious Private Set Intersection." In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2020, Zagreb, Croatia, May 10-14, 2020*. Vol. 12106. Lecture Notes in Computer Science. Springer, 2020, pp. 739–767. DOI: 10.1007/978-3-030-45724-2_25 (cit. on p. 84).

[251] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. "Phasing: Private Set Intersection Using Permutation-Based Hashing." In: *Proceedings of the 24th USENIX Security Symposium, USENIX Security 2015, Washington, D.C., USA, August 12-14, 2015*. USENIX Association, 2015, pp. 515–530. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/pinkas (cit. on p. 73).

[252] Benny Pinkas, Thomas Schneider, Oleksandr Tkachenko, and Avishay Yanai. "Efficient Circuit-Based PSI with Linear Communication." In: *Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2019, Darmstadt, Germany, May 19-23, 2019*. Vol. 11478. Lecture Notes in Computer Science. Springer, 2019, pp. 122–153. DOI: 10.1007/978-3-030-17659-4_5 (cit. on pp. 72–75, 79, 82, 84).

[253] Benny Pinkas, Thomas Schneider, Christian Weinert, and Udi Wieder. "Efficient Circuit-Based PSI via Cuckoo Hashing." In: *Proceedings of the 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2018, Tel Aviv, Israel, April 29 - May 3, 2018*. Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 125–157. DOI: 10.1007/978-3-319-78372-7_5 (cit. on p. 73).

[254] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. "The Loopix Anonymity System." In: *Proceedings of the 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. USENIX Association, 2017, pp. 1199–1216. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/piotrowska (cit. on p. 30).

[255] Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. "Understanding Hierarchical Methods for Differentially Private Histograms." In: *Proceedings of the 39th International Conference on Very Large Data Bases, VLDB 2013, Riva del Garda, Trento, Italy, August 26 - 30, 2013*. Vol. 6. 14. 2013, pp. 1954–1965. DOI: 10.14778/2556549.2556576 (cit. on pp. 140, 145).

[256] Aswin N. Raghavan, Harini Ananthapadmanaban, Manimaran Sivasamy Sivamurugan, and Balaraman Ravindran. "Accurate Mobile Robot Localization in Indoor Environments Using Bluetooth." In: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, May 3-7, 2010*. IEEE, 2010, pp. 4391–4396. DOI: 10.1109/ROBOT.2010.5509232 (cit. on p. 13).

[257] Mishaal Rahman. *Here Are the Countries Using Google and Apple's Covid-19 Contact Tracing API*. XDA. Accessed: 2024-01-16. 2021. URL: https://www.salute.gov.it/portale/nuovocoronaviru

s/dettaglioNotizieNuovoCoronavirus.jsp?lingua=italiano&menu=notizie&p=dalministero&id=4849 (cit. on p. 23).

[258] Ramesh Raskar et al. "Comparing Manual Contact Tracing and Digital Contact Advice." In: *Computing Research Repository (CoRR)* abs/2008.07325 (2020). URL: https://arxiv.org/abs/2008.07325 (cit. on p. 10).

[259] Reuters. *Chinese Officials Punished for Changing Health Codes of Bank Depositors - State Media*. Accessed: 2024-02-15. 2022. URL: https://www.reuters.com/article/idUSL4N2YA03D/ (cit. on p. 43).

[260] Reuters. *German Restaurants Object After Police Use Covid Data for Crime-Fighting*. Accessed: 2024-02-06. 2020. URL: https://www.reuters.com/article/idUSKCN24W2K6/ (cit. on pp. 3, 43).

[261] Reuters. *South Korea Scrambles to Contain Nightclub Coronavirus Outbreak*. The Straits Times. Accessed: 2024-02-20. 2020. URL: https://www.straitstimes.com/asia/east-asia/south-korea-scrambles-to-contain-new-coronavirus-outbreak-threatening-seoul (cit. on pp. 3, 12).

[262] Mohamed Er Rida, Fuqiang Liu, Yassine Jadi, Amgad Ali Abdullah Algawhari, and Ahmed Askourih. "Indoor Location Position Based on Bluetooth Signal Strength." In: *Proceedings of the Second International Conference on Information Science and Control Engineering, ICISCE 2015, Changsha, China, 18 - 20 December 2020*. IEEE, 2015, pp. 769–773. DOI: 10.1109/ICISCE.2015.177 (cit. on p. 14).

[263] Peter Rindal and Mike Rosulek. "Malicious-Secure Private Set Intersection via Dual Execution." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 1229–1242. DOI: 10.1145/3133956.3134044 (cit. on p. 84).

[264] Peter Rindal and Phillipp Schoppmann. "VOLE-PSI: Fast OPRF and Circuit-PSI from Vector-OLE." In: *Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2021, Zagreb, Croatia, October 17-21, 2021*. Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 901–930. DOI: 10.1007/978-3-030-77886-6_31 (cit. on p. 87).

[265] Ronald L. Rivest, Adi Shamir, and Yael Tauman. "How to Leak a Secret." In: *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2001, Gold Coast, Australia, December 9-13, 2001*. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 552–565. DOI: 10.1007/3-540-45682-1_32 (cit. on p. 57).

[266] Ronald L. Rivest et al. *PACT: Private Automated Contact Tracing*. Accessed: 2024-01-11. 2020. URL: https://pact.mit.edu/wp-content/uploads/2020/11/The-PACT-protocol-specification-2020.pdf (cit. on pp. 10, 18, 23, 28, 29, 38, 164, 166).

[267] Robert Koch Institute. *Coronavirus Disease 2019 (Covid-19) - Daily Situation Report of the Robert Koch Institute*. Accessed: 2024-01-12. 2020. URL: https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Dez_2020/2020-12-22-en.pdf?__blob=publicationFile (cit. on p. 85).

[268] Josyl Mariela B. Rocamora and Jhoanna Rhodette Pedrasa. "Evaluation of Hierarchical DHTs to Mitigate Churn Effects in Mobile Networks." In: *Computer Communications* 85 (2016), pp. 41–57. DOI: 10.1016/J.COMCOM.2016.02.003 (cit. on p. 53).

[269] Pablo Rodríguez et al. "A Population-Based Controlled Experiment Assessing the Epidemiological Impact of Digital Contact Tracing." In: *Nature Communications* 12.1 (2021), p. 587. DOI: 10.1038/s41467-020-20817-6 (cit. on p. 41).

[270] Philipp Roos. *No Personal Data, no Food? The New German Covid-19 Regulations and Their Data-Protection Relevance for the Food and Drink Industry*. Freshfields Bruckhaus Deringer. Accessed: 2024-02-09. 2020. URL: https://digital.freshfields.com/post/102g7db/no-personal-data-no-food-the-new-german-covid-19-regulations-and-their-data-pro (cit. on pp. 42, 100).

[271] Jilian A Sacks et al. "Introduction of Mobile Health Tools to Support Ebola Surveillance and Contact Tracing in Guinea." In: *Global Health: Science and Practice* 3.4 (2015), pp. 646–659. DOI: 10.9745/GHSP-D-15-00207 (cit. on pp. 10, 11).

[272] Muhammad Usama Sardar, Saidgani Musaev, and Christof Fetzer. "Demystifying Attestation in Intel Trust Domain Extensions via Formal Verification." In: *IEEE Access* 9 (2021), pp. 83067–83079. DOI: 10.1109/ACCESS.2021.3087421 (cit. on pp. 129, 130).

[273] Sanjay Sareen, Sandeep K. Sood, and Sunil Kumar Gupta. "IoT-Based Cloud Framework to Control Ebola Virus Outbreak." In: *Journal of Ambient Intelligence and Humanized Computing* 9.3 (2018), pp. 459–476. DOI: 10.1007/s12652-016-0427-7 (cit. on p. 11).

[274] Sajin Sasy, Sergey Gorbunov, and Christopher W. Fletcher. "ZeroTrace: Oblivious Memory Primitives from Intel SGX." In: *Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018. URL: https://www.ndss-

symposium.org/wp-content/uploads/2018/02/ndss2018%5C_02B-4%5C_Sasy%5C_paper.pdf (cit. on p. 133).

[275] Felix Sattler et al. "Risk Estimation of SARS-CoV-2 Transmission from Bluetooth Low Energy Measurements." In: *NPJ Digital Medicine* 3.1 (2020), p. 129. DOI: 10.1038/s41746-020-00340-0 (cit. on p. 15).

[276] Hannah Schmitz, Carol L Howe, David G Armstrong, and Vignesh Subbian. "Leveraging Mobile Health Applications for Biomedical Research and Citizen Science: A Scoping Review." In: *Journal of the American Medical Informatics Association (JAMIA)* 25.12 (2018), pp. 1685–1695. DOI: 10.1093/jamia/ocy130 (cit. on pp. 108, 109, 112, 114).

[277] James Scott, Jon Crowcroft, Pan Hui, and Christophe Diot. "Haggle: A Networking Architecture Designed around Mobile Users." In: *Proceedings of the Third Annual Conference on Wireless On-demand Network Systems and Services, WONS 2006, Les Ménuires, France, 18-20 January, 2006*. IFIP, 2006, pp. 78–86. URL: https://2006.wons-conference.org/program.html (cit. on p. 11).

[278] LH Segura Anaya, Abeer Alsadoon, Nectar Costadopoulos, and PWC Prasad. "Ethical Implications of User Perceptions of Wearable Devices." In: *Science and Engineering Ethics* 24 (2018), pp. 1–28. DOI: 10.1007/s11948-017-9872-8 (cit. on p. 114).

[279] Alexander Seifert and Corneel Vandelanotte. "The Use of Wearables and Health Apps and the Willingness to Share Self-Collected Data among Older Adults." In: *Aging and Health Research* 1.3 (2021), p. 100032. DOI: 10.1016/j.ahr.2021.100032 (cit. on p. 108).

[280] Alexander Senger. "Contact Tracing for Super Spreader Events through Active Lighthouses on Top of GAEN." Master thesis. Humboldt University Berlin, 2021 (cit. on pp. 95, 100).

[281] Henning Silber et al. "Linking Surveys and Digital Trace Data: Insights from Two Studies on Determinants of Data Sharing Behaviour." In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 185.S2 (2022), S387–S407. DOI: 10.1111/rssa.12954 (cit. on p. 114).

[282] Silicon Laboratories. *Bluetooth Advertising Data Basics*. Accessed: 2024-01-22. 2024. URL: https://docs.silabs.com/bluetooth/4.0/general/adv-and-scanning/bluetooth-adv-data-basics (cit. on p. 51).

[283] Lucy Simko et al. "Covid-19 Contact Tracing and Privacy: A Longitudinal Study of Public Opinion." In: *Digital Threats: Research and Practice (DTRAP)* 3.3 (2022), pp. 1–36. DOI: 10.1145/3480464 (cit. on pp. 39, 40, 42, 43).

[284] Selena Simmons-Duffin and Robert Stein. *CDC Director: 'Very Aggressive' Contact Tracing Needed for U.S. to Return to Normal*. npr. Accessed: 2024-01-11. 2020. URL: https://www.npr.org/sections/health-shots/2020/04/10/831200054/cdc-director-very-aggressive-contact-tracing-needed-for-u-s-to-return-to-normal (cit. on p. 9).

[285] Laurent Simon, David Chisnall, and Ross J. Anderson. "What You Get is What You C: Controlling Side Effects in Mainstream C Compilers." In: *Proceedings of the IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*. IEEE, 2018, pp. 1–15. DOI: 10.1109/EUROSP.2018.00009 (cit. on p. 137).

[286] Singapore Government Developer Portal. *SafeEntry – National Digital Check-in System*. Accessed: 2024-02-09. 2024. URL: https://www.developer.tech.gov.sg/products/categories/digital-solutions-to-address-covid-19/safeentry/overview.html (cit. on pp. 42, 90).

[287] Singapore Government Developer Portal. *TraceTogether*. Accessed: 2024-02-06. 2023. URL: https://www.developer.tech.gov.sg/products/categories/digital-solutions-to-address-covid-19/tracetogether/overview.html (cit. on pp. 11, 18).

[288] Eleanor Singer. "Exploring the Meaning of Consent: Participation in Research and Beliefs about Risks and Benefits." In: *Journal of Official Statistics* 19.3 (2003), p. 273. URL: https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/exploring-the-meaning-of-consent-participation-in-research-and-beliefs-about-risks-and-benefits.pdf (cit. on p. 114).

[289] Natasha Singer and Choe Sang-Hun. *As Coronavirus Surveillance Escalates, Personal Privacy Plummets*. New York Times. Accessed: 2024-01-11. 2020. URL: https://www.nytimes.com/2020/03/23/technology/coronavirus-surveillance-tracking-privacy.html (cit. on pp. 3, 9, 43).

[290] Anya Skatova and James Goulding. "Psychology of Personal Data Donation." In: *PLOS ONE* 14.11 (2019). DOI: 10.1371/journal.pone.0224240 (cit. on p. 113).

[291] Nigel P. Smart. *Cryptography Made Simple*. Information Security and Cryptography. Switzerland: Springer, 2016. ISBN: 3319219359 (cit. on pp. 29, 34, 64, 133).

[292] Emil Stefanov et al. "Path ORAM: An Extremely Simple Oblivious RAM Protocol." In: *Journal of the ACM* 65.4 (2018), pp. 1–26. DOI: 10.1145/3177872 (cit. on pp. 68, 134, 171).

[293] Ion Stoica, Robert Tappan Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications." In: *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 2001, San Diego, CA, USA, August 27-31, 2001*. ACM, 2001, pp. 149–160. DOI: 10.1145/383059.383071 (cit. on pp. 53, 54).

[294] Naomi F Sugie. "Utilizing Smartphones to Study Disadvantaged and Hard-To-Reach Groups." In: *Sociological Methods & Research* 47.3 (2018), pp. 458–491. DOI: 10.1177/0049124115626176 (cit. on p. 113).

[295] Mihaly Sulyok and Mark Walker. "Community Movement and Covid-19: A Global Study Using Google's Community Mobility Reports." In: *Epidemiology & Infection* 148 (2020). DOI: 10.1017/S0950268820002757 (cit. on p. 127).

[296] Chanjuan Sun and Zhiqiang Zhai. "The Efficacy of Social Distance and Ventilation Effectiveness in Preventing Covid-19 Transmission." In: *Sustainable Cities and Society* 62 (2020), p. 102390. DOI: 10.1016/j.scs.2020.102390 (cit. on p. 16).

[297] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. "Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12." In: *Computing Research Repository (CoRR)* abs/1709.02753 (2017). URL: http://arxiv.org/abs/1709.02753 (cit. on p. 147).

[298] Martin Thomson and Christopher A. Wood. *Oblivious HTTP*. Tech. rep. Work in Progress. Internet Engineering Task Force, Aug. 2023. 49 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-ohai-ohttp/10/ (cit. on p. 125).

[299] Eran Toch and Oshrat Ayalon. "How Mass Surveillance Crowds Out Installations of Covid-19 Contact Tracing Applications." In: *Proceedings of the ACM on Human-Computer Interaction* 7.CSCW1 (2023), pp. 1–26. DOI: 10.1145/3579491 (cit. on pp. 31, 39, 40).

[300] Tor Blog. *Domain Fronting Is Critical to the Open Web*. Accessed: 2023-03-03. 2018. URL: https://blog.torproject.org/domain-fronting-critical-open-web (cit. on p. 123).

[301] Tor Project. *TOR Project*. Accessed: 2024-01-26. 2022. URL: https://www.torproject.org (cit. on pp. 53, 96, 123, 125, 126).

[302] Simon Trang, Manuel Trenz, Welf H. Weiger, Monideepa Tarafdar, and Christy M. K. Cheung. "One App to Trace Them All? Examining App Specifications for Mass Acceptance of Contact-Tracing Apps." In: *European Journal of Information Systems* 29.4 (2020), pp. 415–428. DOI: 10.1080/0960085X.2020.1784046 (cit. on p. 39).

[303] Max Tretter. "Sovereignty in the Digital and Contact Tracing Apps." In: *Digital Society* 2.1 (2023), p. 2. DOI: 10.1007/s44206-022-00030-2 (cit. on p. 23).

[304] Ni Trieu, Kareem Shehata, Prateek Saxena, Reza Shokri, and Dawn Song. "Epione: Lightweight Contact Tracing with Strong Privacy." In: *IEEE Data Engineering Bulletin* 43.2 (2020), pp. 95–107. URL: http://sites.computer.org/debull/A20june/issue1.htm (cit. on pp. 18, 20, 29, 36–38, 66, 72, 85, 163, 166).

[305] Carmela Troncoso et al. "Decentralized Privacy-Preserving Proximity Tracing." In: *IEEE Data Engineering Bulletin* 43.2 (2020), pp. 36–66. URL: http://sites.computer.org/debull/A20june/issue1.htm (cit. on pp. 15, 18, 23, 28, 31, 33, 34, 38, 46, 71, 77, 85, 86, 93, 164, 166).

[306] Jason Uher, Ryan G. Mennecke, and Bassam S. Farroha. "Denial of Sleep Attacks in Bluetooth Low Energy Wireless Sensor Networks." In: *Proceedings of the IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, November 1-3, 2016*. IEEE, 2016, pp. 1231–1236. DOI: 10.1109/MILCOM.2016.7795499 (cit. on p. 32).

[307] Nik Unger et al. "SoK: Secure Messaging." In: *Proceedings of the IEEE Symposium on Security and Privacy, S&P 2015, San Jose, CA, USA, May 17-21, 2015*. IEEE Computer Society, 2015, pp. 232–249. DOI: 10.1109/SP.2015.22 (cit. on pp. 94, 123, 124).

[308] United Nations. *Universal Declaration of Human Rights*. Accessed: 2024-02-27. 1948. URL: https://www.un.org/en/about-us/universal-declaration-of-human-rights (cit. on p. 2).

[309] UpGuard, Inc. *14 Biggest Healthcare Data Breaches*. Accessed: 2023-03-06. 2023. URL: https://www.upguard.com/blog/biggest-data-breaches-in-healthcare (cit. on p. 108).

[310] Serge Vaudenay. "Analysis of DP3T." In: *IACR Cryptology ePrint Archive* (2020), p. 399. URL: https://eprint.iacr.org/2020/399 (cit. on pp. 33, 38).

[311] Serge Vaudenay. "Centralized or Decentralized? The Contact Tracing Dilemma." In: *IACR Cryptology ePrint Archive* (2020), p. 531. URL: https://eprint.iacr.org/2020/531 (cit. on p. 20).

[312] Michael Veale. *Privacy Is Not the Problem with the Apple-Google Contact-Tracing Toolkit*. The Guardian. Accessed: 2024-02-15. 2020. URL: https://www.theguardian.com/commentisfree/2020/jul/01/apple-google-contact-tracing-app-tech-giant-digital-rights (cit. on p. 43).

[313] Liang Wang et al. "Inference of Person-To-Person Transmission of Covid-19 Reveals Hidden Super-Spreading Events during the Early Outbreak Phase." In: *Nature Communications* 11.1 (2020), p. 5006. DOI: 10.1038/s41467-020-18836-4 (cit. on p. 42).

[314] Xiao Shaun Wang et al. "Oblivious Data Structures." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS 2014, Scottsdale, AZ, USA, November 3-7, 2014*. ACM, 2014, pp. 215–226. DOI: 10.1145/2660267.2660314 (cit. on pp. 127, 131, 133, 138, 170).

[315] Glenn Webb et al. "A Model of the 2014 Ebola Epidemic in West Africa with Contact Tracing." In: *PLOS Currents* 7 (2015). DOI: 10.1371/currents.outbreaks.846b2a31ef37018b7d1126a9c8adf22a (cit. on p. 10).

[316] Royce J. Wilson et al. "Differentially Private SQL with Bounded User Contribution." In: *Proceedings on Privacy Enhancing Technologies, PETs 2020, Online, July 14–18, 2020*. 2020. DOI: 10.2478/popets-2020-0025 (cit. on p. 147).

[317] Chris Wymant et al. "The Epidemiological Impact of the NHS Covid-19 App." In: *Nature* 594.7863 (2021), pp. 408–412. DOI: 10.1038/s41586-021-03606-z (cit. on p. 41).

[318] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks." In: *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2005, Urbana-Champaign, IL, USA, May 25-27, 2005*. ACM, 2005, pp. 46–57. DOI: 10.1145/1062689.1062697 (cit. on p. 32).

[319] Jason Xue. *Who's at Fault in Celebrity Health Code Hack?* TechNode. Accessed: 2024-02-15. 2021. URL: https://technode.com/2021/01/11/whos-at-fault-in-celebrity-health-code-hack/ (cit. on p. 43).

[320] Eiko Yoneki. "FluPhone Study: Virtual Disease Spread Using Haggle." In: *Proceedings of the 6th ACM Workshop on Challenged Networks, CHANTS@MOBICOM 2011, Las Vegas, NV, USA, September 19-23, 2011*. ACM, 2011, pp. 65–66. DOI: 10.1145/2030652.2030672 (cit. on p. 10).

[321] Xingliang Yuan, Xinyu Wang, Cong Wang, Chenyun Yu, and Sarana Nutanong. "Privacy-Preserving Similarity Joins Over Encrypted Data." In: *IEEE Transactions on Information Forensics and Security* 12.11 (2017), pp. 2763–2775. DOI: 10.1109/TIFS.2017.2721221 (cit. on pp. 138, 147).

[322] Kuan Zhang, Xiaohui Liang, Jianbing Ni, Kan Yang, and Xuemin Sherman Shen. "Exploiting Social Network to Enhance Human-To-Human Infection Analysis without Privacy Leakage." In: *IEEE Transactions on Dependable and Secure Computing* 15.4 (2018), pp. 607–620. DOI: 10.1109/TDSC.2016.2626288 (cit. on p. 11).

[323] Zhaoyang Zhang, Honggang Wang, Xiaodong Lin, Hua Fang, and Dong Xuan. "Effective Epidemic Control and Source Tracing through Mobile Social Sensing Over WBANs." In: *Proceedings of the IEEE International Conference on Computer Communications 2013, INFOCOM 2013, Turin, Italy, April 14-19, 2013*. IEEE, 2013, pp. 300–304. DOI: 10.1109/INFCOM.2013.6566783 (cit. on p. 11).

[324] Wenting Zheng et al. "Opaque: An Oblivious and Encrypted Distributed Analytics Platform." In: *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*. USENIX Association, 2017, pp. 283–298. URL: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/zheng (cit. on p. 151).