Department of Computer Science
Series of Publications A
Report A-2022-7

Privacy-Preserving Protocols for Protected Networking

Sara Ramezanian

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki in Auditorium CK112, Exactum, on the 10th of June, 2022 at 12 o'clock.

University of Helsinki
Finland

Supervisors
Valtteri Niemi, University of Helsinki, Finland
Tommi Meskanen, University of Helsinki, Finland

Pre-examiners
Isaac Agudo, University of Malaga, Spain
Tobias Pulls, Karlstad University, Sweden

Opponent
Panos Papadimitratos, KTH Royal Institute of Technology, Sweden

Custos
Valtteri Niemi, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645 (print)
ISSN 2814-4031 (online)
ISBN 978-951-51-8206-7 (paperback)
ISBN 978-951-51-8207-4 (PDF)
Helsinki 2022
Unigrafia

Privacy-Preserving Protocols for Protected Networking

Sara Ramezanian

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
sara.ramezanian@cs.helsinki.fi
http://cs.helsinki.fi/Sara.Ramezanian/

PhD Thesis, Series of Publications A, Report A-2022-7
Helsinki, May 2022, 88 + 113 pages
ISSN 1238-8645 (print)
ISSN 2814-4031 (online)
ISBN 978-951-51-8206-7 (paperback)
ISBN 978-951-51-8207-4 (PDF)

Abstract

Digital technologies have become an essential part of our lives. In many parts of the world, activities such as socializing, providing health care, leisure and education are entirely or partially relying on the internet. Moreover, the COVID-19 world pandemic has also contributed significantly to our dependency on the on-line world.

While the advancement of the internet brings many advantages, there are also disadvantages such as potential loss of privacy and security. While the users enjoy surfing on the web, service providers may collect a variety of information about their users, such as the users' location, gender, and religion. Moreover, the attackers may try to violate the users' security, for example, by infecting the users' devices with malware.

In this PhD dissertation, to provide means to protect networking we propose several privacy-preserving protocols. Our protocols empower internet users to get a variety of services, while at the same time ensuring users' privacy and security in the digital world. In other words, we design our protocols such that the users only share the amount of information with the service providers that is absolutely necessary to gain the service. Moreover, our protocols only add minimal additional time and communication costs, while leveraging cryptographic schemes to ensure users' privacy and security.

The dissertation contains two main themes of protocols: privacy-preserving set operations and privacy-preserving graph queries. These protocols can be applied to a variety of application areas. We delve deeper into three application areas: privacy-preserving technologies for malware protection, protection of remote access, and protecting minors.

Computing Reviews (2012) Categories and Subject Descriptors:

Privacy → Privacy enhancing technologies → Cryptography
Multi-party compitation → Private Set Operations

General Terms:
Cryptography, Privacy enhancing technologies, Secure multi-party computations

Additional Key Words and Phrases:
Edge computing, cloud computing, 5G Networks, AI

Acknowledgements

Without a doubt my journey to write PhD thesis has been the most rewarding time of my life, and for that I am deeply indebted to many people. Foremost, I am extremely grateful to my supervisors, Prof. Valtteri Niemi, and Dr. Tommi Meskanen. This endeavor would not have been possible without their unconditional support, immense knowledge, patience, and insightful comments. Their kind supervision allowed my studies to go the extra mile. It has been indeed a special privilege and a pleasure to work with them and learn from them. I sincerely hope that I have a chance to collaborate with them, also in the future.

I would like to extend my gratitude to my amazing co-authors; Prof. Valtteri Niemi, Dr. Tommi Meskanen, Prof. Ville Junnila, Prof. Jian Liu, and Mr. Masoud Naderpour. I could not have been able to complete this research without their invaluable help. I am also deeply thankful to Dr. Pirjo Moen, Prof. Sasu Tarkoma, Prof. Jussi Kangasharju, and Dr. Kimmo Järvinen, for their helpful comments and advices throughout my doctoral studies. I am sincerely grateful to Prof. Isaac Agudo and Dr. Tobias Pulls, for their precious time and effort to pre-examine this thesis.

My sincere thanks also goes to staff of the Department of Computer Science, especially Mr. Pekka Niklander, and Ms. Minna Lauri, who have always offered me their help and support. I am also thankful to my many friends at the department, for making my life full of joy and happiness.

I am also grateful to Business Finland, European Union's Horizon 2020 Research and Innovation Program, and Academy of Finland for providing the funding that supported this research. Thanks should also go to Nokia Foundation for granting me the Nokia Scholarship in 2019.

Special thanks to the board of Doctoral School in Computer Science (DoCS) for accepting me as a member of the board for two years (2018-2019). I am also grateful to the PhD students of the department, who have trusted me to be their PhD representative at the board of DoCS. I also thank F-Secure Corporation, especially Mr. Alexey Kirichenko and Mr. Paolo Palumbo, for our research collaboration, which led to a patent.

Last but not least, I owe my utmost heartfelt gratitude to my beloved family. A family like mine is a treasure greater than any I can imagine. My precious Dad, my dearest Mom, and my amazing Brother, words cannot express how incredibly fortunate I feel to have you in my life. Thank you for nourishing my soul so I can be victorious in life. You are the most valuable blessing of my life. Thank you for being my best friend, and biggest cheerleader. You always support and encourage me to be my true-self, and are the proudest souls of my achievements. My gratitude to you is infinite and endless. Thank you. Thank you. Thank you.

This research is on topics that make human digital lives safer, and hopefully, make the world a better place. This is what the thesis is dedicated to.

Helsinki, May 2022
Sara Ramezanian

Contents

List of Included Articles

I Tommi Meskanen, Jian Liu, Sara Ramezanian, and Valtteri Niemi. "Private membership test for Bloom filters." In *Proceedings of IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 515-522. IEEE, 2015.

II Sara Ramezanian, Tommi Meskanen, Masoud Naderpour, Ville Junnila, and Valtteri Niemi. "Private membership test protocol with low communication complexity." In *Digital Communications and Networks*, vol. 6, no. 3, pp. 321-332. 2020.

III Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "Privacy preserving queries on directed graph." (Short paper) In *Proceedings of the 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-5. IEEE, 2018.

IV Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "Privacy-Protecting Algorithms for Digraph Shortest Path Queries." In *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 10, no. 3, pp. 86-100. 2019.

V Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "Privacy preserving 2-party queries on bipartite graphs with private set intersection." (Short paper) In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1867-1870. 2019.

VI Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "AI-based Cyberbullying Prevention in 5G networks." In *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 11, no. 4, pp. 1-20. 2020.

VII Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "Parental Control with Edge Computing and 5G Networks" In *Proceedings of the 29th Conference of Open Innovations Association (FRUCT)*, pp. 290-300. IEEE, 2021.

VIII Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. "Multi-party Private Set Operations with an External Decider" In *Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy (DBSec)*, pp. 117-135. Springer, 2021.

This PhD dissertation is based on the above eight peer-reviewed publications. Publications II, IV and VI are extended versions of earlier published conference papers by Ramezanian et al. [1, 2, 3], respectively.

Author's contributions

I The present author evaluated the performances of the proposed protocols in this publication. The co-authors together did the literature review and development of the protocols. The present author also participated in the writing of performance evaluations, and contributed in preparing the presentation of the publication. The present author's Master's thesis, published in 2016 at University of Turku, Finland, was partially based on this publication.

II The present author initially proposed the idea behind the protocol. The authors all together contributed to the literature review, design, implementation and analysis of the protocol. The present author wrote the manuscript while the co-authors gave important contributions in reviewing the manuscript.

III The present author performed the literature review, implemented the protocol, wrote the manuscript, and prepared the presentation of this publication. All the authors contributed in planning the publication, analyzing the data, designing the protocol, and analyzing the protocol.

IV All the authors participated in generating the initial idea, planning the publication, literature review, analyzing the data, designing and analyzing the protocol. The present author wrote the manuscript and implemented the protocol. The co-authors provided significant contributions in reviewing the manuscript.

V All the authors contributed to planning the publication, literature review, designing and analyzing the protocol. The present author had a leading role in writing the manuscript while the co-authors gave important contributions in reviewing the manuscript. The present author prepared the poster presentation for the publication.

VI The present author was in the lead of the planning of the publication, literature review, writing the manuscript, and implementing the pro-

tocol. The present author also proposed the original idea of a privacy-preserving protocol for the purpose of cyberbullying prevention. The authors all together further developed the idea behind the protocol into a detailed scheme, and contributed to analyzing the protocol. The co-authors gave significant contributions in reviewing the manuscript.

VII The present author was in the lead of planning the publication, literature review, and writing the manuscript. The present author proposed the idea of utilizing a centralized parental control system that uses edge computing and performs in a privacy-preserving way. The present author analyzed the protocol, prepared the presentation of the publication and carried out the implementation parts. The co-authors provided important contributions at all stages of performing research and preparing the manuscript.

VIII The present author proposed the original idea behind the protocol. The authors together expanded the original idea into a more detailed scheme. The co-authors significantly contributed in making the protocols more efficient. All the authors performed an extensive literature review, and contributed in analyzing the protocol, and preparing the presentation of the publication. The present author had a leading role in the writing process while the co-authors gave important contributions in reviewing the manuscript.

Chapter 1

Introduction

We live in an era in which it is hard to avoid using the internet in everyday life. There are more and more devices that can connect to the internet, and there is an increase in the services that are run via the internet. Video conferencing, internet banking, virtual learning environments, and on-line shopping are examples of services that are provided over the internet.

The advancement of the internet is a double-edged sword: it brings many advantages but it has also introduced new challenges and risks to our lives [4]. For instance, social media can have a negative impact on the academic performance of its users [5] and their self-esteem [6], it can also cause false medical self-diagnosis [7]. Among other challenges and risks, utilizing the internet may cause threats to the users' privacy and security [8, 9].

Informally speaking, *privacy* in the digital world is a state in which a person can perform an on-line task without exposing anything unwanted about themself. *Security* in the digital world specifies the circumstances in which a person, or an object, is protected from harm that may occur while using the internet.

Usually people have a tendency to sacrifice privacy and security to get better access to on-line services [10, 11]. On the other hand, service providers may collect information about their clients, and further investigate this information. For instance, just a snapshot of applications that a user has installed on their phone can reveal sensitive information such as the user's religion, ethnicity, place of residence, and sexual orientation [12].

Network security is a well-known concept in Computer Science. Roughly speaking, a computer network is a group of computers that are communicating with each other and using shared resources [13]. Network security is sometimes seen as a subset of computer security, and refers to techniques and rules that are designed to protect the security of a network [14]. In this

dissertation, we are not only looking to protect the network. Instead, we are looking at the protection from a wider perspective, such that it includes the network, its users, and other devices that try to build a network.

Protected networking refers to procedures that ensure the protection of both information and users. For instance, protecting systems against malicious software and unwanted access, and protecting users from malicious digital contents such as bullying.

In this dissertation we propose several privacy-preserving methods for the purpose of protected networking. In other words, we present methods that enable users to access a variety of on-line services, without losing their privacy and security. At the core of each of our methods there is a *protocol*.

A protocol involves two or more parties. The parties exchange information based on a set of rules, in such a way that at the end of the protocol they can achieve a common goal. In a privacy-preserving protocol, each party only sends the amount of information that they want to share, and nothing more. In order to design a privacy-preserving protocol, we use *cryptography*.

Cryptography [15] is a field of study that is designed to protect information from being exposed to an adversary, by using several scientific disciplines such as mathematics and computer science. The intention of adding cryptographic protocols to a system is to make it more protected. However, these protocols may require adding extra rounds of computations and communications between parties.

We aim to create cryptographic protocols such that internet users can get their requested services in a private manner, without sacrificing time and bandwidth. In other words, utilizing our protocols implies only minor time and communication costs.

In some of our protocols, we use *distributed computing* [16]. Simply put, distributed computing refers to a model where part/all of the computation that needs to be done in a protocol is distributed over two or more computing devices that communicate over a network.

We apply our protocols to real-life scenarios, and specifically we study three application areas: i) malware protection, ii) protection of remote access, and iii) protecting minors. Now, we briefly describe these three application areas; why they are important, and why privacy matters in their settings.

Malware Protection: A piece of malware can be described as a piece of software that can perform harmful acts on a computer. If malware finds its way to a computer the system would be infected. Just like humans, an infected computer may spread malware to other computers that are

communicating with this infected computer. Therefore, computers should be protected against malware infections and an infected system should also be protected against the malware that is in the system.

We now give an example to show the importance of malware protection in a private way. Let us assume that a user has a file that might be infected by a piece of malware, and there is a security company that performs malware checking. The user wants to make sure whether the file is clean or not, without disclosing the content of the file to the company. If the result of malware checking shows that the file is indeed infected, the user gives the computer to the company for clean-up. The company also wants to know which malicious files are spreading. In this example, the user wants to perform malware checking in a private manner. One trivial way to perform the malware checking in a privacy-preserving way is to give the set of all known malware samples to the user. Then the user can look for a specific malware sample without disclosing any information to anybody else. However, this approach has several shortcomings. For instance, it may cause the malware samples to spread, and it requires heavy usage of bandwidth. Therefore, we want to design secure and efficient protocols that provide malware protection in a privacy-preserving way.

Protection of Remote Access: A user has *remote access* privileges if they are authorized to access a computer or a set of computers that are not physically close to the user via internet [17]. If user A in one computer has permission to remotely access another computer as a user B, we say there is a *trust relation* from A to B. User A of computer 1 can access user B of computer 2, if A is authorized to do so by a system administrator, and B can authenticate that it is actually user A who wants to access B. Typically, the authentication is implemented by using a secret key that only user A has. We detail the process to issue the trust relation from A to B, in Chapter 4.

When there is a trust relation from user A to B, the digital materials and resources of user B are accessible to user A as well. Therefore, the process to check whether or not to grant such an access is critical from a security and privacy point of view, and consequently, managing trust relations are important to protect computers.

If user A can access user B, and user B can access user C, then user A can also access user C. In other words, trust relations could be in a chain. Let us assume a scenario where user A is compromised and therefore the secret key is revealed to an adversary. We also assume that user C learns that user A is compromised. User C wants to know whether the adversary (who now can act as if he/she is user A), can access C through some

intermediary users. However, the intermediary users (e.g., user B) do not want to reveal which users they have or do not have access to, and whether or not they have been compromised. In this example, user C needs to learn the intermediary user(s) that make it possible for A to indirectly access C so that C can block access from that intermediary user(s). However, C should not learn the other users who do not cause this indirect access, but can access via A. Therefore, we want to design query protocols to privately search in the trust relations. After C has checked the indirect access from A to C, if there is an intermediary user that directs the compromised user to C, it should be revealed.

Protecting Minors: As already mentioned, the rise of digitalization may cause harm to its users. Minors are the most vulnerable type of internet users, therefore they should be protected against potential digital harms. On the other hand, just like any other type of internet users, minors have the right to privacy.

One of the potential harms of on-line socializing is cyberbullying. Before the globalization of internet, a bully only had accessed to his/her victim while they were in the same location, e.g., during school time. Nowadays, a bully may have access to his/her victim twenty-four hours a day, seven days a week by using the internet. One way to protect minors from bullies is to check the content of the digital messages that a minor receives. However, the personal messages may be privacy sensitive. Also, most of the messages are benign and are not intended to be harmful.

In addition to cyberbullying, there are other potential threats towards minors, such as websites with adult contents and videos with violent scenes. Protection against these harmful materials also require monitoring the child's on-line activities. However, just like every other user, the child's activities on the internet are privacy sensitive. Therefore, we aim to protect minors in the internet by developing privacy-preserving protocols.

This doctoral dissertation is based on eight original peer-reviewed publications (that are listed on pages ix and x). Publication I and Publication II present protocols to enable privacy-preserving look-up for a certain item, for example, a malware sample. Publication III, Publication IV, and Publication V propose privacy-preserving protocols that enable queries on trust relational databases, or any database that is similar in structure to trust relations. Publication VI and Publication VII provide multiple privacy-preserving minor protection protocols. The protocols of Publication VI and Publication VII can also be applied to other user-protection settings than minors, for example, cyberbullying protection for celebrities. Finally, Publication VIII presents several protocols that can be applied to a variety

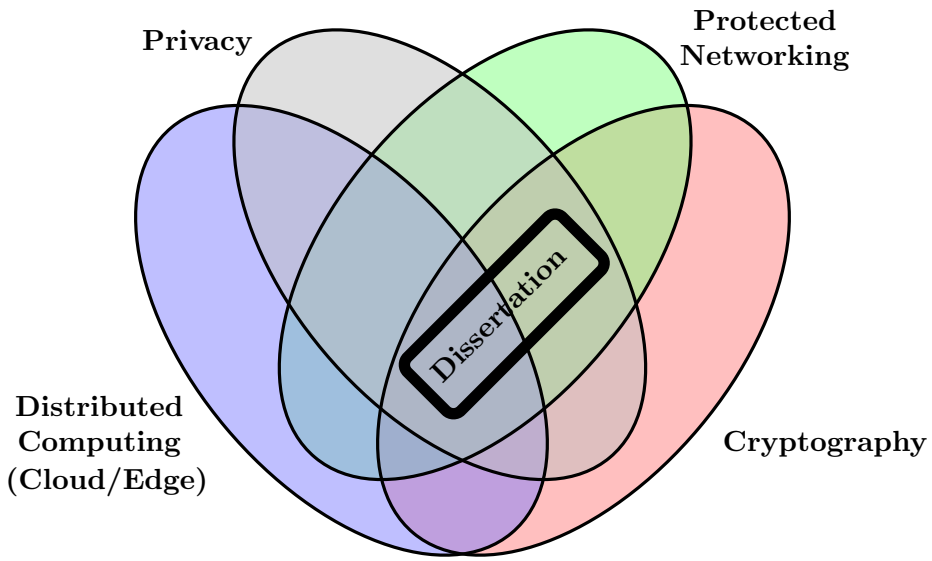


Figure 1.1: Positioning of this dissertation in the field.

of application areas, including the three areas that we described in this chapter.

In a nutshell, this dissertation provides several novel and efficient privacy-preserving protocols, and studies three application areas that these protocols can be applied to. We test the efficiency of our protocols by running multiple experiences in realistic settings, and we show that our protocols can be used in practice. A visualization of the main areas that the dissertation falls under and the positioning of this work in these fields is presented in Figure 1.1.

The rest of the dissertation is structured as follows: Chapter 2 provides the definitions of several concepts that are frequently used in this dissertation. In Chapter 3, we give the technical preliminaries that are important for understanding the subsequent chapters. Our research questions and methodologies are provided in Chapter 4. Moreover, we give our detailed contributions in this chapter, as well. We propose several novel privacy-preserving protocols in Chapter 5. Then, in Chapter 6 we delve into three application areas for privacy-preserving protected networking. Finally, we conclude this dissertation in Chapter 7.

Chapter 2

Background

This chapter gives the necessary background on the concepts that are used in the dissertation. We begin with a closer look at the notions of security, privacy, and protected networking. Then, we give definitions of the data structures that we choose to store the data in. Then, we give brief introductions to Artificial Intelligence and 5G Networks and their components that we use to design the findings of this dissertation.

2.1 Security and Privacy

As it is defined in the Oxford dictionary, *security* has several meanings and use-cases [18]. Security can refer to a person, an organization, or a valuable item that is used as protection against potential future threats. Security can also refer to a state of mind that is free from worry, or to describe the assurance that something valuable is present (e.g., food security, and job security). In this dissertation when we talk about security, we only consider the security of *computer information systems*[1].

A computer information system is an information system[2] that utilizes a computer to perform all or part of its assigned tasks [20]. The components of a computer information system are i) hardware, ii) software, iii) data (or databases), iv) people[3], and v) procedures that instruct how the previous four components are interacting. The security of *computer information systems* refers to practice of protecting the above five components from

[1] Also called security of computer-based information systems.

[2] An information system is designed to collect and store information, and contains methods to further process and distribute this information [19].

[3] Here, "people" refers to 1) staff (people who are working on the information system) and 2) users (people who are customers of the information system).

harm [21]. For instance, in order to protect the security of data, we should maintain its confidentiality, integrity, and availability [22].

In article "*Conceptualizing Privacy*", Solove [23] surveyed the definitions for privacy, and categorized the cores to conceptualize privacy to six groups: i) the right to be let alone, ii) the ability to limit access to the self, iii) the action of secrecy, iv) the ability to control personal information, v) the protection of one's person-hood, and vi) the control over one's intimate aspects of life. Solove then concluded that the attempts to unify the definition for privacy have not been satisfying. Other scholars also expressed difficulties in conceptualizing the notion of privacy, such as Gutwirth [24], Sieghart [25], and Bennett [26].

Although a general definition of privacy is out of reach, categorizing the types of privacy can help identify the potential privacy issues that may occur. In 2013, Finn et al. [27] presented seven types of privacy: Privacy of i) person, ii) behavior and action, iii) communication, iv) data and image, v) thoughts and feelings, vi) location and space, and vii) associations.

The "right to privacy" [28] may have a different meaning for each individual [29]. Although most people have the same expectation when it comes to security, not everyone shares the same privacy concerns.

Privacy-preserving protocols refer to protocols that enable users to utilize digital communication networks without sacrificing their privacy. In this dissertation, we provide specific privacy goals for each protocol separately and therefore, there is no need to have a single general definition for privacy.

2.2 Data Structure

A data structure is a concept that can be used as a tool to store, manage and organize information [30]. In computer science, a structure is qualified as a data structure if it consists of two components: i) a set of data with a collection of relationships among them, and ii) a function (or a collection of functions) which can be utilized with the data [31]. Bloom filter, Cuckoo filter and various types of graphs are a few examples of data structures that are widely used in theory and in practice. We now describe the data structures used in the dissertation.

2.2.1 Bloom Filter

In 1970, Bloom [32] introduced a probabilistic data structure to perform membership tests on big databases. The space-efficiency of the Bloom filters make them a popular data structure in real-life applications.

Since 1970, several variants of Bloom filters have been proposed in academia [33]. However, in this work we use the original filter that was proposed by Bloom [32]. This filter works as follows:

A Bloom filter is an array of m bits and initially all the bits are set to be zero. In order to insert the elements of a set X into the filter, we require l independent hash functions H_i, $i = 1, ..., l$, where the outputs of these hash functions are mapped uniformly to set $M = \{1, 2, ..., m\}$. In order to insert the set X into the filter, we feed each element x of the set X to all the hash functions. Since each output of a hash function $H_i(x)$ is in the set M, each hash value can be considered as an array position. Consequently, for each $x \in X$ we obtain l pseudorandom array positions, and we set the value of those positions in the Bloom filter to 1. Now, the Bloom filter represents the set X.

Naturally, to query an element y from the Bloom filter one should obtain all the outputs of l hash functions and check the values of the corresponding positions of the filter. If the outputs are all pointing to the positions with value 1, then y is in the filter (or in the set that the filter is representing), and otherwise, y is not in the filter.

A query from the Bloom filter never results in a false negative, however, it might result in a false positive. This is due to the fact that all the hash values that are representing an element which has never been added to the filter, might point to positions of the array that are set to 1 because of other elements. If a Bloom filter represents a set X with n elements, then the probability of a false positive query from this filter is $(1 - e^{-ln/m})^l$.

2.2.2 Cuckoo Filter

Although the simple construction of the Bloom filter makes it a good candidate to store and manage data, a search for an even more space-efficient data structure has been ongoing. In 2014, Fan et al. [34] proposed a data structure that is more space-efficient than a Bloom filter. Their proposed data structure is also probabilistic and is called a Cuckoo filter.

Each cuckoo filter is an array of *buckets*. Each bucket has b *positions*. Let us assume that we want to insert a set X to a Cuckoo filter. To do so, we first need to obtain the *fingerprints* of all elements $x \in X$. The fingerprint of x is a short bit string of length f that is computed by using a hash function.

Now, the fingerprint of x is inserted into the Cuckoo filter, by using another hash function *hash*. This fingerprint can be stored in one of the

two buckets $h_1(x)$ or $h_2(x)$ calculated as follows:

$$\begin{cases} h_1(x) = hash(x) \\ h_2(x) = h_1(x) \oplus hash(x\text{'s fingerprint}). \end{cases}$$

The choice between the two candidate buckets is done based on which one has free space. If they both have free space, Cuckoo filter chooses one of the buckets randomly. If the buckets are both full, Cuckoo filter chooses one of the buckets randomly. Let us call this bucket i. Now, the Cuckoo filter tries to relocate one of the fingerprints in bucket i to its other candidate bucket (bucket j), which is $j = i \oplus hash(\text{fingerprint})$. If bucket j is also full, the Cuckoo filter repeats the relocating procedure for one of the fingerprints in j. If the relocating fails for 500 consecutive times, the cuckoo filter is considered to be full and it is not possible to store another fingerprint in it. Each Cuckoo filter has a parameter α, the load factor, which indicates how full the filter is.

As with Bloom filters, a query from a Cuckoo filter may result in a false positive but never in a false negative. False positive happens when an item that has not been added to the filter has the same fingerprint (and therefore, same value for function $hash$) as an item that is already in the filter. An upper bound for the false positive rate of a Cuckoo filter is $\epsilon \leq 1 - (1 - 1/2^f)^{2b}$. In order to achieve this upper bound, the fingerprints should have a minimum size of $\lceil \log_2(2b/\epsilon) \rceil$ bits, which is equal to $\lceil \log_2(1/\epsilon) + \log_2(2b) \rceil$ bits.

2.2.3 Graph

Graphs are data structures that are widely used in many different fields such as telecommunications and chemistry. In this subsection, we give the formal definition of a graph, then we define two variants of graphs that are used in our studies: *Directed Graph* and *Bipartite Graph*.

A graph G consists of a set of finite number of vertices, V, and a set of E which is the set of edges that connect pairs of vertices. In other words, each edge is a pair $\{u, v\}$, and shows that there is a relation between vertices u and v [35]. The illustration of a graph is done by drawing the vertices of V as points, and joining them with lines, where the two ends of each line are the pair $\{u, v\}$ in the set E. If the order of vertices in the pair $\{u, v\}$ is relevant, then the graph is a directed graph (digraph). In other words, the set of edges in a digraph is a collection of ordered pairs (u, v). In that case, to illustrate G one should draw an arrow from u to v.

A bipartite graph is a graph that contains two disjoint independent sets of vertices V_1 and V_2. In the bipartite graph, each edge $\{u, v\}$ connects

these two sets of vertices, i.e., $u \in V_1$ and $v \in V_2$, or vice versa. A bipartite graph can be an undirected or directed graph.

2.3 Artificial Intelligence

The term Artificial Intelligence (AI) is used with various meanings by computer science experts and non-experts [36]. However, AI is commonly referred to as demonstrating problem solving and learning[4] that is performed by a machine [38]. Russell and Norvig [39] defined AI as "the designing and building of intelligent agents that receive percepts from the environment and take actions that affect that environment".

Artificial intelligence is a field of computer science that has a variety of applications. Here we only briefly give definitions to concepts and problems that we use in our protocols.

Machine Learning is a sub-field of AI that studies methods to build algorithms that make computers able to improve in performing a certain task by experience, and not by being explicitly programmed to do this task [40]. In other words, a machine can improve its performance for any given task, by only running a machine learning algorithm and learning from experiences.

A *classifier* is an algorithm that sorts the (unlabeled) data to (labelled) categories. These categories are called *classes*, and the entire process is called *classification*. The term classification was first used in statistics, and in 1936 Fisher [41] delivered the first method for statistical classification. We can use machine learning for the purpose of classification [42].

Classifiers themselves are often categorized as *Binary Classifiers* or *Multi-class Classifiers*. A binary classifier [43] categorizes the data to two classes, for instance in quality control of products, as "accepted" or "failed". A multi-class classifier [44] sorts the data into multiple categories, e.g., classifying fruits into several predefined classes.

Natural language processing (NLP) [45] is a sub-field of AI that develops algorithms for computers to "understand" human languages. In other words, the computer can recognize the context and style of a text, such as sarcasm, humor, hate, and not just the words.

[4]The ability to learn and problem solving are amongst attributes that are considered as having "intelligence" [37].

2.4 Distributed Computing Paradigm

A distributed computing system is a combination of two or more components[5] that can communicate with each other either via a local network or internet. These components can cooperate in such a way that they can carry out a common task [46]. Now, we define two paradigms of distributed computing: cloud computing and edge computing.

Cloud Computing

The term cloud computing was used by Chellappa [47] in 1997 for the first time. Cloud computing refers to utilizing several combined technologies, such as storage, processing powers, virtualization, to deliver computing services over the internet. As each person/organization only pays for the service they obtain from the cloud, the cloud-based services are cost-efficient. Speed in computing, high performance and agility in accessing resources are some of the benefits of utilizing a cloud.

Edge Computing

Edge computing refers to a paradigm of a distributed computing system that aims to perform data computations close to where the data has been generated, i.e., at the edge of the network [48].

With the increase of telecommunication technologies, data proliferation has become a concern. Although cloud computing helps in dealing with data processing in a fast and cost-efficient way, some of the new technologies (such as IoT[6]) require a real-time response that is not possible to achieve by using a cloud. The architecture of edge computing consists of technologies that enable data processing to be executed at the edge of a network. The main benefits of utilizing an edge computing system are low latency, privacy, scalability, and cost efficiency.

2.5 5G Networks

A mobile network is a telecommunication network where the connections to and from the nodes are wireless. The mobile networks were established in the late 1970s to cover voice calls and were utilizing analog signals to

[5] Please note that in principle these components can also work alone, and independently from each other.

[6] Internet of Things, e.g., smart refrigerators, smart watches, and heat sensors.

transmit radio communication [49]. From the second generations of mobile networks (2G) onwards, radio signals are digital.

The 5G networks are the fifth generation of the mobile networks. 5G is evolved from its predecessor, LTE/4G. Compared to 4G, the 5G networks aim to deliver a significantly improved connectivity technology to cover the increasing need to move and process the data via cellular networks. The advantages of 5G over its predecessor are massive connectivity, higher capacity, low latency, reduced energy consumptions, and very high throughput (1-20 Gbps) [50].

The system architecture of 5G networks is designed in such a way that it is easier to add new modules to the system than in, e.g., 4G networks. In this dissertation, we use this feature of 5G to enhance the privacy and security of the users.

The architecture of 5G, its components and their functionalities are explained in 3GPP[7] specifications [51]. Here, we briefly describe two of the functions that are implemented in the 5G networks. We utilize these functions later in this dissertation.

User Plane Function

One of the main functions of 5G is User Plane Function (UPF), and it is a component of the core network. The UPF is responsible for handling the traffic from the User Equipment (UE) by performing several tasks such as packet routing/forwarding and packet inspection.

Policy Control Function

The policy rules that are required to handle each subscriber are provided by Policy Control Function (PCF). Other functions of 5G networks can connect to PCF to obtain the correct policy for a certain user and therefore, can provide suitable services for that user.

[7] 3GPP is a consortium that consists of organizations that develop standardized protocols for telecommunication networks.

Chapter 3

Technical Preliminaries

In this chapter we present the necessary definitions on the technical preliminaries that are used later in this dissertation.

3.1 Secure Multi-party Computation

A secure multi-party computation (MPC) is a cryptographic protocol between n parties. Each party i holds a secret input x_i, and has no information about the other parties' inputs. The parties want to jointly compute the output of a function f over their secret inputs. The goal of a secure MPC protocol is to compute and reveal to each party the value of $f(x_1, x_2, ..., x_n)$ without revealing any information about the secret inputs, other than what can be deduced from the output alone.

In 1982, Yao [52] proposed the first secure 2-party computation protocol to solve the millionaires' problem[1]. The topic of secure MPC and its application areas have been studied extensively [53, 54, 55].

3.2 Private Set Operation

In this dissertation we consider *set operation* as a set intersection, set union, set complement or a combination of these operations. We use the notation $\bar{S}_i$ for the complement of set S_i.

It is known that the output set S_T of any set operation from input sets $S_1, ..., S_n$ can be written in a *Disjunctive Normal Form* (DNF):

$$S_T = (A_{1,1} \cap ... \cap A_{1,\alpha_1}) \cup ... \cup (A_{\beta,1} \cap ... \cap A_{\beta,\alpha_\beta}) \tag{3.1}$$

[1]The millionaires' problem refers to a setting in which two parties want to know which one of two is more wealthy without revealing their actual wealth to each other, or any third party.

where $A_{i,j} \in \{S_1, ..., S_n, \bar{S}_1, ..., \bar{S}_n\}$, and $1 \leq \alpha \leq n$ and $\beta \in \mathbb{N}$. Please note that for each input set, at most one of the two sets S_i and $\bar{S}_i$ appears in every intersection of Equation 3.1.

Alternatively, for a different α and β the output set S_T of any set operation can be written in a *Conjunctive Normal Form* (CNF):

$$S_T = (A_{1,1} \cup ... \cup A_{1,\alpha_1}) \cap ... \cap (A_{\beta,1} \cup ... \cup A_{\beta,\alpha_\beta}) \cdot \tag{3.2}$$

Please note that for each input set, at most one of the two sets S_i and $\bar{S}_i$ appears in every union of Equation 3.2.

Please also note that set inclusion and set equality can also be considered as set operations. However, in this dissertation we do not consider these operations, because output of set inclusion and set equality is not a set.

A Private Set Operation (PSO) is a cryptographic protocol that involves at least two parties. Each party has a private input set and the parties all together want to perform a set operation on the input sets. At the end of the protocol the parties will only learn the output set and nothing else.

PSO is a subset of secure MPC that has been studied separately. The wide applicability of the variants of PSO encourages the researchers to study this problem in a variety of settings.

Private Set Intersection (PSI) is a variant of PSO where the outcome of the protocol is the intersection of the input sets. See for example [56]. PSI is extensively studied separately.

Private Membership Test (PMT) is a special variant of 2-party PSI, for example, see [57]. In a PMT protocol, one of the parties (Alice) has a private item x and the other party (Bob) has a private set X. Alice wants to privately learn whether x is in Bob's set. Bob also wants to keep his set private.

3.3 Homomorphic Encryption

A cryptosystem is homomorphic if the system permits computations on ciphertexts that correspond to computations with the plaintexts, without having to decrypt the ciphertexts first [58]. There are three types of homomorphic encryption: partially homomorphic, somewhat homomorphic, and fully homomorphic encryption schemes [59]. In this dissertation we only discuss partially homomorphic encryption schemes. If a cryptosystem is partially homomorphic, it either supports multiplication or addition on the ciphertexts, but not both. For instance, if $E_k(x)$ is an encryption func-

tion of an additively homomorphic cryptosystem with a key k, and every plaintext x is in a set M, then for some operation $\odot$:

$$E_k(a) \odot E_k(b) = E_k(a+b) \quad \text{for all} \quad a, b \in M\cdot \tag{3.3}$$

In Equation 3.3 the operation on the left-hand side does not have to be the addition.

3.3.1 Goldwasser-Micali Homomorphic Encryption

In 1982, Goldwasser and Micali [60] proposed the first provably-secure probabilistic encryption scheme. Their scheme is additively homomorphic. In this cryptosystem the public key is a pair (N, y), where N is the product of two large and distinct prime numbers p and q, y is a quadratic non-residue modulo N and $Jacobi(y, N) = 1$. The private key is the pair (p, q). The encryption function Enc to encrypt a bit x is $Enc(x)$, and the output of this function is $r^2 y^x$, where r is a random number in $\mathbb{Z}_N^*$. Therefore, the ciphertext c is $r^2 y^x$.

In order to decrypt c, we need to find out whether c is a *quadratic residue* modulo N. An integer c is a quadratic residue modulo N if there is an integer a such that the following equation holds:

$$a^2 \equiv c \mod N\cdot \tag{3.4}$$

The ciphertext c decrypts to 0 if it is a quadratic residue, and otherwise it decrypts to 1.

3.3.2 Paillier's Cryptosystem

One of the well-known additively homomorphic encryption schemes is the Paillier cryptosystem that is proposed by Paillier and Pointcheval [61]. This public-key cryptosystem is probabilistic. The public key is a pair (N, g), where $N = pq$, and p and q are two distinct safe primes[2] of the same size, and $g \in \mathbb{Z}_{N^2}^*$ such that N divides the order[3] of g.

The private keys are p, q, and $\lambda = lcm(p-1, q-1)$. In order to encrypt a plaintext $x \in \mathbb{Z}_N$, we need to pick a random number d that belongs to $\mathbb{Z}_N^*$. The Paillier encryption function $E_g(x, d)$ is equal to $g^x d^N \mod N^2$. The ciphertext w is equal to $g^x d^N \mod N^2$ and belongs to $\mathbb{Z}_{N^2}^*$.

[2] A prime Q is called a safe prime if a prime number P exists such that $Q = 2P + 1$.

[3] The order of element g in a cyclic group G is the number of elements in G that are generated by g. A cyclic group is a group that can be generated by one element.

Let us use the notation $L(u)$ to show a function that outputs $(u-1)/N$ for $u \in \mathbb{Z}^*_{N^2}$ and $u-1$ that is divisible by N. To decrypt w we use the following equation:

$$x = D_g(w) = \frac{L(w^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} \mod N. \tag{3.5}$$

3.4 Private Information Retrieval

In 1995, Chor et al. [62] introduced the problem of Private Information Retrieval (PIR). This cryptographic concept is studied extensively [63, 64, 65]. There are two parties in a single-server PIR protocol; Alice who has a database X with n items and Bob who has an index i where $1 \leq i \leq n$. Bob wants to privately obtain the value that is stored in the ith index of X without revealing his index i. Alice does not require that her database is kept private but she hopes to find a way to provide the response to Bob's query in a non-trivial way (i.e., not sending the whole database to Bob). One example of a single-server PIR protocol is presented by Chang [66].

Chang's PIR Scheme

The PIR protocol by Chang [66] consists of two parties; a server that has a 2-dimensional database $X_{h\times h}$ and a client who has an index (i^*, j^*). The client wants to retrieve the value $x(i^*, j^*)$ which is the element of X that is located at index (i^*, j^*). Firstly, by using the encryption function E_g from Paillier's cryptosystem, the client computes vectors α and β, with h components, as follows:

$$\alpha_t = E_g(I(t, i^*), r_t) \text{ and } \beta_t = E_g(I(t, j^*), s_t), \tag{3.6}$$

where $t \in \{1, 2, ..., h\}$, random numbers r_t and s_t, are chosen uniformly from $\mathbb{Z}^*_N$ and I is the identity matrix

$$I(t, t') = \begin{cases} 1 & \text{if } t = t' \\ 0 & \text{otherwise.} \end{cases}$$

Please note that when the components of vectors α and β are decrypted, all other components will decrypt to zero except α_{i^*} and β_{j^*}, which will decrypt to one.

Server receives vectors α and β from the client, and computes a vector σ with h components:

$$\sigma_i = \prod_{t=1}^{h} (\beta_t)^{x(i,t)} \mod N^2 \tag{3.7}$$

where $i = 1, 2, ..., h$. Note that because of the homomorphic properties of Paillier's cryptosystem, the decryption of each σ_i is equal to:

$$D_g(\sigma_i) = \sum_{t=1}^{h} (\beta_t) x(i,t) \mod N^2. \tag{3.8}$$

The server is required to continue similar computations as Equation 3.7 with the components of vectors β and σ. However, the components of vector σ have the same bit-size of N^2. Therefore, the server computes $u_i, v_i \in \mathbb{Z}_N$ in such a way that $\sigma_i = u_i N + v_i$. Finally the server computes results u and v as follows:

$$u = \prod_{t=1}^{h} (\alpha_t)^{u_t} \mod N^2 \text{ and } v = \prod_{t=1}^{h} (\alpha_t)^{v_t} \mod N^2 \tag{3.9}$$

and sends these results to the client. The client computes:

$$x(i^*, j^*) = D_g(D_g(u)N + D_g(v)), \tag{3.10}$$

and therefore, retrieves the element of X in position (i^*, j^*), in a privacy-preserving manner.

3.5 Oblivious Pseudorandom Function

Before presenting the definition of an *Oblivious Pseudorandom Function*, we define some terminology that is needed for this definition.

An *efficiently computable function* is a function that operates with an efficient amount of computational power (such as time and space), and produces the desired output. A function is said to be polynomial time computable if the function runs and produces the output with computational resources bounded by a polynomial size of the input [67].

A *Random oracle* is a function that generates truly random outputs that are distributed uniformly on the output range [68].

A *Pseudorandom Function (PRF)* f is a keyed function, with a key k. The function f is efficiently computable and its output $f_k(\cdot)$ is indistinguishable[4] from the output of a random oracle [69].

An Oblivious Pseudorandom Function (OPRF) [70] is defined as a cryptographic protocol to compute the output of a PRF f. This protocol involves two parties: a sender S with a key k, and a receiver R with an

[4]Informally speaking, here "indistinguishable" means that the two outputs are similar in every aspect and therefore, it is not possible to recognize one output from the other.

input x. After executing the protocol R can compute $f_k(x)$ without revealing any information to S. In particular, the sender S does not learn any information about the output $f_k(x)$ or the input x.

3.6 Cryptographic Hash Function

A hash function is an efficiently computable function and therefore, has the property of mapping the input data of arbitrary size to a fixed-length bit string output. These outputs are called hash values or message digests.

A *cryptographic hash function* [71] is a deterministic[5] hash function h such that: i) h is quick and easy to compute, ii) the output is uniformly distributed over a finite set, iii) even a small change in the input data results in a completely different hash value, iv) h is *Collision resistant*; i.e., it is infeasible to find two different inputs that result in the same hash value, v) h is *Preimage-resistant*; i.e., by having a hash value y it is computationally infeasible to find the input data that maps to y, and vi) h is *2nd-preimage resistant*; i.e., by having a certain input data x it is not possible to find another input x' that maps to the same hash value, i.e., $h(x) = h(x')$.

3.7 Blind Signature

A digital signature scheme is used to preserve the message from being altered and to show the recipient that the message is authentic and is indeed generated by the claimed sender.

In 1983, Chaum [72] proposed the concept of *blind signature* to enable signing the digital messages without knowing their content. More precisely, a blind signature is a cryptographic protocol that has two parties; Alice who possesses a message m and Bob who possesses a key k (the signing key). After executing the protocol, Alice receives the signature of m without learning any information about k, and Bob does not learn anything about the message m.

3.8 Adversarial Model

In cryptography, an adversary is a party that is either part of the system (an insider entity) or is outside the system, and who has a malicious intention such as learning other parties' secret input, corrupting the system or

[5]Feeding the same input data to the function always results into the same message digest.

eavesdropping [73]. The traditional adversarial model is called *Dolev-Yao* [74]. A Dolev-Yao intruder is assumed to be the strongest model for an outsider attacker [75]. A Dolev-Yao intruder can access and intercept any messages and replies that are sent between parties. This model is considered to be too restrictive [76], because in most protocols an adversarial model that can also consider the participants of the protocol as potential attackers, is needed. Although extremely skillfull, a Dolev-Yao intruder is rather infeasible [77], and therefore in this dissertation we consider a more realistic way to describe the adversarial models: *semi-honest (passive)* and *malicious (active)* adversaries [76].

3.8.1 Semi-honest

In the semi-honest adversarial model, the parties follow the protocol as it is prescribed. However, the parties try to collect and save the messages that they received from other parties. Then, they try to extract more data from these messages than the protocol is designed to give. For instance, in a PSO protocol, a semi-honest adversary sees messages that are sent by all parties and may try to analyse these messages to learn the size of their sets [78].

3.8.2 Malicious

A malicious adversary is assumed to have any arbitrary computational functionalities that are polynomial-time computable. A malicious entity does not follow the protocol as it is transcribed, and might try to perform harmful activities such as completely demolish the system, change the inputs or outputs, or any other deviation from the protocol, to gather private information about other parties. For example, in a PSO protocol, a malicious party should play their part either with their set A or its complement $\bar{A}$, but this malicious party always uses A. Therefore, the output set would not be the actual result of the set operation [78].

Chapter 4

Research Questions and Methodologies

In this chapter we present the main research questions that this dissertation focuses on. Then, we describe the methodologies that are used to answer these research questions.

4.1 Research Questions

We formulate some of the existing challenges in protected networking into the following two main research questions:

RQ 1: How do we design methods for utilizing the benefits of the digital world without sacrificing privacy?

RQ 2: How do we protect privacy while using a distributed computing system?

In order to research RQ 1 and RQ 2, we look into three tools; cryptography, 5G networks, and data structures, which are commonly used in many protocols. We want to utilize the benefits of these tools while enhancing them with privacy-protection techniques. In this regard, we aim to answer the following research questions:

RQ 3: How do we apply cryptography to ensure privacy?

RQ 4: How do we utilize 5G networks to protect privacy?

RQ 5: How do we apply privacy-preserving methods with different data structures?

As we mentioned before, PSO protocols have a variety of applications. Although the design of PSO protocols is studied extensively, the existing solutions are constructed to compute a specific set operation. Therefore, we want to investigate the following research question:

RQ 6: How do we develop protocols that enable any privacy-preserving set operation?

In the dissertation we study the problem of privacy-preserving protected networking specifically in three application areas: i) malware protection, ii) protection of remote access, and iii) protecting minors.

For each application area, we first give a realistic scenario. Then, we formulate a specific set of privacy goals and detailed research questions for each of the application areas.

Malware Protection

Malware is built with the intention of performing malicious acts on the infected device. Enormous efforts have been devoted to designing malware-checking systems [79]. In this dissertation we aim to create techniques protecting against malware such that the techniques are also privacy-preserving.

Scenario (i): A server $\mathcal{S}$ has a database of known malware samples. A client $\mathcal{C}$ is interested in installing an application x, and wants to see whether this application is malware-free or not. The server does not want to reveal its malware samples to the client or anybody else. The client also does not want to reveal x to the server (unless x turned out to be malware) or anybody else. The server also wishes to use a cloud-assisted solution to enable queries on its database.

Goal (i): To design a privacy-preserving protocol that enables the client to privately query for x in the malware database that is stored in a cloud.

RQ i-1: How do we design a protocol for Goal (i) that minimizes the time complexity of this privacy-preserving query?

RQ i-2: How do we design a protocol for Goal (i) that minimizes the communication cost of this privacy-enhancing technique?

Protection of Remote Access

In order to control whether a remote access should be allowed, manage the remote access, and protect it from intruders, we first need to store the remote access rules in a suitable format. One way to store the remote access rules is to insert each one of the rules in a quintuple (source-user, source-host, fingerprint, target-user, target-host). In this dissertation, a *trust relation* refers to a rule granting permission for a user at a host (source-user at source-host) to access another user at another host (target-user at target-host) via an authentication key that has a unique fingerprint. Therefore, the above quintuple represents a trust relation. In Chapter 6 we

show that we can draw a graph that represents a trust relational database. Please note that the trust relations can be in a chain, such that if user A has remote access to B, and B can access C, then A has remote access to C. Therefore, a user can be a source-user in one trust relation and target-user in another trust relation.

Scenario (ii): A database owner has a database of trust relations, which are quintuples that are of form (source-user, source-host, fingerprint, target-user, target-host).

Goal (ii): The owner wants to enable the admins of different hosts to perform queries on the database in a privacy-protecting way.

RQ ii-1: Is it possible to know whether there is a trust relation between user A and B, in a privacy-preserving way?

RQ ii-2: If there is a trust between two users, is it possible to retrieve the intermediary nodes (users) in a privacy-preserving way?

RQ ii-3: Let us assume that there are two separate trust relations databases that are managed by two different administrators. Is it possible to determine whether there is a trust relation between source-user A in one database to target-user B in the other database, in a privacy-preserving way?

Protecting Minors

Everyday, more and more digital technologies are developed. Nowadays, life without utilizing the internet is almost impossible. There are several challenges that the rise of digitalization can bring for its users. For instance, challenges in digitalized healthcare systems [80], in education [81], and in public transport [82]. In this dissertation, we specifically focus on children (minors), who are the most vulnerable types of users in the digital world.

Scenario (iii): A minor wants to use the internet. We assume that the parents of this minor are not monitoring their child's on-line activity all the time.

Goal (iii): To design techniques that make the digital world a more secure place for minors, while at the same time these techniques preserve the minor's privacy.

RQ iii-1: Can a minor be automatically protected against cyberbullying in a privacy-preserving way?

RQ iii-2: Can a minor be protected from potentially harmful content of certain web pages by automatic means in a privacy-preserving way?

RQ iii-3: Can a minor be automatically protected from potentially harmful text messages in a privacy-preserving way?

RQ iii-4: Can parents influence their children's on-line activity while the child's privacy and the parent's privacy are preserved against each other and also the network?

In Chapters 5 and 6, we investigate answers to all the research questions of this chapter. Table 4.1 presents the mapping of publications to both sets of research questions and application areas.

4.2 Methodologies

Research methodology is a procedure that defines a phased program to conduct research [83]. The research methodologies are discipline specific, i.e., each field of research requires a specific set of research methods. Bailey [84] proposed a two-step research methodology for the discipline of security science; a pilot study followed by a case study.

Regardless of which discipline the conducted research is related to, a literature review can be one of the research methodologies to give us the state of the art in our field of research [85]. In addition to a literature review, controlled experiments, ethnography and action research are other research methodologies that can be used, e.g., in the software developing field [86].

The nature of the research problem has direct influence on the choice between different research methodologies [87]. In this work, we want to enhance privacy-preserving technologies and therefore, we choose the following seven research methodologies that contributes to our nature of research. In each of the original publications of this dissertation (Publication I - Publication VIII), we employed all these methods, but to a different extent. The chosen methodologies are:

- **Background research:** Foremost, we conduct a comprehensive survey-like research on the topic. We then recognize the state of the art on that topic, and list the most relevant and significant works in our publications.

- **Identifying challenges:** We identify challenges in the existing solutions. Specifically, we identify how privacy is violated in those solutions.

- **Formation of the problem in the formal/precise model:** In each publication, we present a formal model to describe the research problem(s).

Pub.	RQ	Application Area	RQ for App Area
I	1, 2, 3, 5	Malware Protection	i-1, i-2
II	1, 2, 3, 5	Malware Protection	i-2
III	1, 2, 3, 5	Protection of Remote Access	ii-1
IV	1, 2, 3, 5	Protection of Remote Access	ii-1, ii-2
V	1, 3, 5	Protection of Remote Access	ii-1, ii-3
VI	1, 3, 4	Protecting Minors	iii-1
VII	1, 2, 3, 4	Protecting Minors	iii-2, iii-3, iii-4
VIII	3, 6	Several	Several

Table 4.1: Mapping of publications to the main research questions, application areas, and their research questions.

- **Design solution for each abstract problem:** We then proceed by proposing novel solutions for each formal model. If applicable, we also explore how the abstract problem is applied to each of the application areas i, ii, and iii.
- **Security analysis:** We investigate the level of security in each of our proposed solutions. We usually consider a situation where adversaries are trying to corrupt our system. We then provide a sketch of security proof for each of our protocols.
- **Privacy analysis:** We analyse the privacy for each party that is involved in our protocols.
- **Performance analysis:** We implement the cryptographic parts and estimate the performances of the other components of our protocols. The parameters that we use are realistic. At this stage, we also compare our solutions with prior ones.

4.3 Dissertation Contributions

The goal of this dissertation is to design privacy-preserving protocols for secure networking, with respect to three important application areas. The main contributions of Publication I - Publication VIII can be categorized to three themes, as follows:

Theme A) Malware Protection: Publication I and Publication II present several private membership test protocols using Bloom filters and

Cuckoo filters. In Publication I, we focus on reducing the computation costs, and in Publication II our focus is on the ways to achieve a low bandwidth usage. Protocols of both publications are tested with the setting of a cloud-based malware checking system. However, the findings of these two publications can be applied to any situations where a PMT test is needed and the database can be stored in a Bloom/Cuckoo filter.

Theme B) Protection of Remote Access: We present several privacy-preserving protocols for graph queries in Publication III, Publication IV and Publication V. In these publications we explain how we can construct several types of graphs from databases of trust relations. Therefore, in addition to other use-cases, Publications III - V can be deployed to perform queries on remote access permissions in a private way.

Theme C) Minor Protection: Publication VI presents privacy-preserving methods to prevent cyberbullying in 5G networks. The findings of Publication VII provide privacy-preserving parental control for all children, regardless of whether or not their parents want to be involved in the on-line activities of their children. Lastly, Publication VIII presents general solutions to any PSO problem. We use a protocol of Publication VIII in our parental control system of Publication VII. However, Publication VIII can be used for any settings where a PSO protocol is needed, for example, in the settings of Publication I, Publication II and Publication V.

The key concepts that are used in each publication are shown in Figure 4.1.

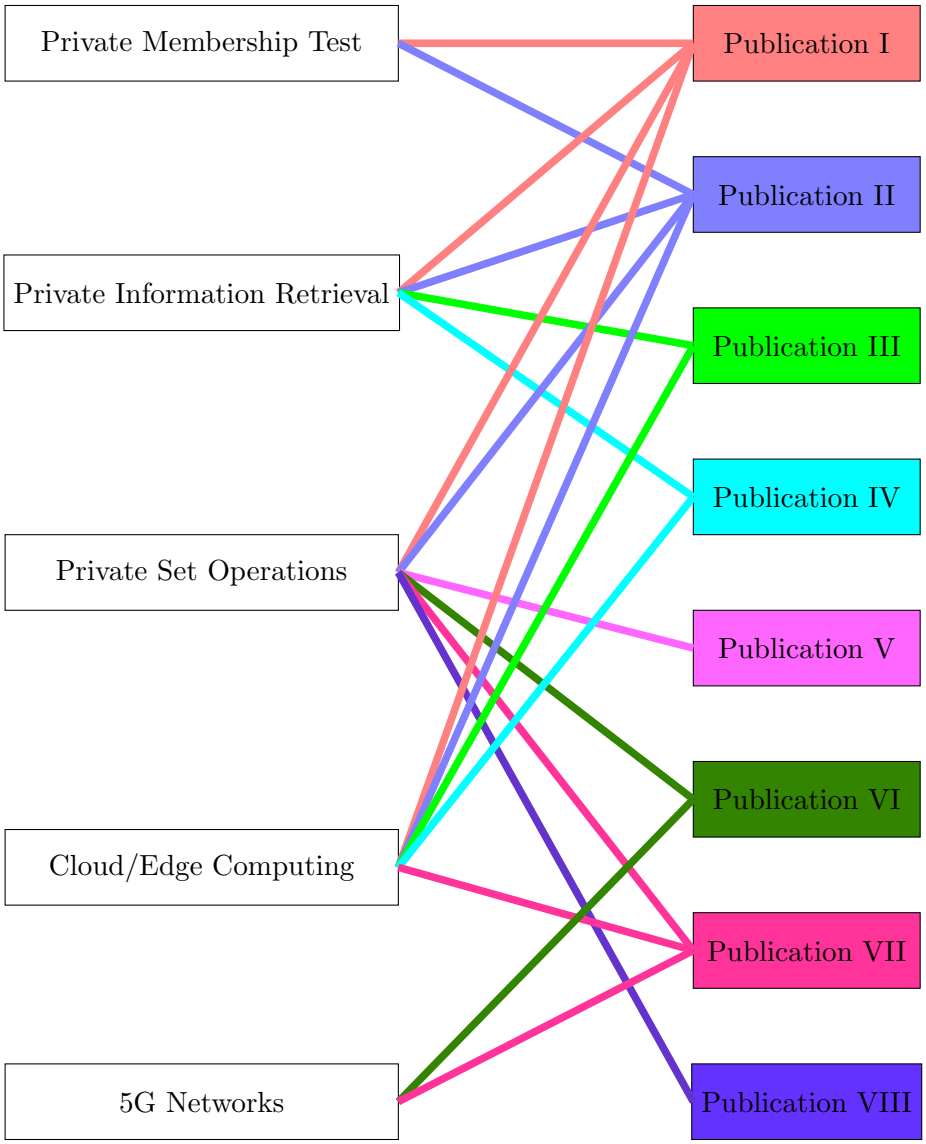


Figure 4.1: Mapping of the key concepts to the publications.

Chapter 5

Privacy-Preserving Protocols

In this chapter, we present the privacy-preserving protocols that we proposed in the publications that are attached to this dissertation.

With the ever increasing need to access data fast and in a secure way, the importance of privacy-enhancing technologies, such as privacy-preserving protocols, is realized now more than ever. These technologies should be designed in such a way that they can be feasible in practice. Among the contributions of this dissertation are the privacy-preserving protocols that have a wide range of applicability, with a sufficiently low execution time that makes them useful in practice. Our proposed protocols can be categorized into two main topics: i) privacy-preserving protocols for set operations and ii) privacy-preserving protocols for graph searches.

5.1 Privacy-Preserving Set Operations

Privacy-preserving set operation protocols have a diverse range of applicability such as genomics applications [88], electronic voting [89], mutual friend/contact discovery in the social media [90], detecting botnets in different internet service providers (ISPs) [91], and speech processing [92].

A special case of PSO problem is private set intersection. There have been many proposals for PSI protocols with different approaches [56, 93, 94], and with different settings [95, 96]. Another special case of PSO problem is private set union, for example, see [97] and [98].

In Section 5.1, we first present protocols for two special PSO problems, other than PSI. Then we present criteria to classify PSO problems.

5.1.1 Private Set Operation with an External Decider

A private set operation with an *external decider* (or a decider D), is a special type of PSO. It can be formalized as follows: There are n parties each with a private input set, and a decider with no input data. The decider wants to learn the outcome of a set operation on the input sets. The parties and the decider interact in such a way that at the end of the protocol the decider learns the outcome without learning anything else about the input sets. In particular, the decider wants to learn one of the following about the output set: 1) whether it is empty, 2) what the cardinality of this output set is, and 3) what the elements in it are.

In order to explain the importance of the setting of a PSO with an external decider, we first present an example of a scenario where it becomes useful. Let us assume that a statistical institute in a country wants to investigate how well the health care system works in that country. As an example, this statistical institute decides to compute the number of citizens who are in the risk group for a certain disease, and who also are covered by an insurance. Therefore, the statistical institute and hospitals together calculate the union of all citizens who have been diagnosed with that certain disease at different hospitals (set A). Then, the statistical institute and insurance companies calculate the union of all citizens who have valid health insurance by different insurance companies (set B). Lastly, the statistical institute potentially together with the insurance companies and the hospitals calculates the cardinality of the set $A \cap B$. As health-related matters are considered to be privacy sensitive, the statistical institute needs to perform these set operations with hospitals and insurance companies in a privacy-preserving manner. The statistical institute wants to carry out the calculations this way to minimize intrusion in individual citizens' lives.

The above scenario is an example of a PSO problem with an external decider (the statistical institute). For further detailed examples of a PSO problem with a decider please see Publication VIII.

Assume that there are n parties in a PSO protocol. Each party P_i has a private set S_i. Let us denote by $\bar{S}_i$ the complement of the set S_i. As we discussed earlier, any set operation can be written in DNF:

$$S_T = (A_{1,1} \cap ... \cap A_{1,\alpha_1}) \cup ... \cup (A_{\beta,1} \cap ... \cap A_{\beta,\alpha_\beta}) \tag{5.1}$$

or alternatively in CNF:

$$S_T = (A_{1,1} \cup ... \cup A_{1,\alpha_1}) \cap ... \cap (A_{\beta,1} \cup ... \cup A_{\beta,\alpha_\beta}) \tag{5.2}$$

where $A_{i,j} \in \{S_1, ..., S_n, \bar{S}_1, ..., \bar{S}_n\}$, $1 \leq \alpha \leq n$ and $\beta \in \mathbb{N}$. The general solution to any PSO problem has been studied before [99, 100], however,

to the best of our knowledge we are the first to propose general solutions that cover all PSO problems in the presence of an external decider[1].

In Publication VIII, we present comprehensive solutions to solve any PSO problems with an external decider, where the universe is limited (hereafter, PSO-Lim problem). We also present a solution to obtain the cardinality of the output set, and whether it is empty, for any PSO where the universe is not limited (hereafter, PSO-UnLim problem). We now briefly explain these solutions.

General Solution to PSO with a Decider for a Limited Universe

We present a general solution to the problem of PSO with a decider for a limited universe (PSO-Lim). We assume that the set operation is already converted to CNF.

When the universe is limited to u items, the universe can be mapped to an ordered set $U = \{a_1, a_2, ..., a_u\}$. The ordering of the elements in this universe is known to all the parties.

In the off-line phase the decider creates public and private keys for a non-deterministic additively homomorphic encryption scheme, and sends the public key to all parties. We assume that the parties together create a shared repository that only they have access to. The decider or anybody else cannot access this repository. The parties together create β vectors W^k for $1 \leq k \leq \beta$, each vector has u components, choose a random number r, compute the encryption of r, and initially set all the components of the vectors W^k to enc(r).

In the on-line phase of our PSO-Lim protocol, the parties modify the vectors W^k to represent the outcome of set operation $(A_{k,1} \cap ... \cap A_{k,\alpha_k})$. To perform this, each party P_i whose input S_i (or $\bar{S}_i$) is one of the sets $\{A_{k,1}, ..., A_{k,\alpha_k}\}$ modifies the vector W^k such that if $a_j \in S_i$ (or $a_j \notin \bar{S}_i$) then party P_i replaces the entry W^k_j with an enc(0). Otherwise, party P_i multiplies the entry W^k_j by an enc(0). Please note that the parties can modify the vector W^k together, and the order of modification does not have an impact on the outcome of the protocol. However, the parties can not modify a specific entry of W^k simultaneously.

After all the vectors W^k are fully modified, one of the parties (e.g., party P_1) computes a new vector Z with u positions, where component j of the vector is $Z_j = \prod_{k=1}^{\beta} W^k_j$ for all $1 \leq j \leq u$.

[1]Feige et al. [101] proposed a protocol to compute AND function in the setting of a secure multi-party computation with the presence of an external decider, however, they did not further develop their protocol to compute any other function than AND.

As we mentioned before, the decider should learn one of the following cases and nothing else: i) what is the output set S_T, ii) what is the cardinality of S_T, or iii) whether S_T is empty. Now, based on which of the above cases needs to be calculated, party P_1 might need to alter Z; if the decider is supposed to learn the whole elements of the output set S_T, party P_1 does not modify the vector Z and sends it to the decider. If the decider requires learning the cardinality of S_T, party P_1 shuffles Z and sends it to D. If the decider is supposed to learn whether S_T is empty or not, party P_1 expands vector Z by adding duplicates of the components of Z to the vector, shuffles it, and sends it to the decider.

Vector Z contains encrypted values, and therefore, the parties cannot retrieve any information about the other parties' input sets from this vector. The decider decrypts Z and learns one of the above three cases as follows: i) for every entry Z_i of Z is decrypted to zero, u_i is in S_T, ii) the number of components of Z that are decrypted to zero, is the cardinality of S_T, or iii) if there are any components of Z that decrypt to zero, then S_T is non-empty, otherwise, S_T is empty.

General Solution to PSO with a Decider for an Unlimited Universe

Utilizing hash values in a PSO setting is one of the naive methods in privacy-preserving set operations protocols [102]. Computing a hash value is significantly faster than computing any ciphertext with a secure public-key cryptosystem. Therefore, using hash values in a PSO protocol makes the protocol more efficient when it is compared to using a public-key cryptosystem to encrypt the private sets. The advantage of using keyed-hash function over simple hashing, lies in the fact that use of the keyed-hash function rules out the possibility of the brute force attack.

We now assume that the set operation is presented in a DNF. In our general solution to PSO with a decider for an unlimited universe (PSO-UnLim) we use keyed-hash functions.

In the off-line phase of our PSO-UnLim protocol, the parties agree on a keyed hash function with a key k. The key k is not revealed to the decider. The parties compute the keyed-hash values of the items in their private sets. The parties together create a big number of dummy hash values[2]. The parties then agree on how to distribute these dummies between themselves such that every possible outcome of a set operation is hidden using these dummies. Therefore, the parties know the number of dummies

[2]Here, a dummy hash value is a random bit string with the size of the output of the keyed hash function.

that corresponds to each possible outcome of the output set. The number of dummy hash values should be significantly bigger than the number of actual hash values.

In the on-line phase of our protocol, based on the set operation that is required by the decider, the involved parties send the real hash values and dummy ones to D. The parties also give the number of dummies that would appear in the output set.

According to the set operation that the decider wants to obtain, D calculates the intersections and unions of the keyed-hash values of the parties, and finds the number of elements in the final set. Then, the decider subtracts the number of dummies from the total number of elements that appeared in the final set, and gets the cardinality of the output set S_T.

As an example, assume there are only 2 parties in the protocols with private sets A and B, and the decider wants to learn the cardinality of $A \cap B$. Assume that the real cardinality is 3, and also assume that A and B agreed to create 12 different dummies that they both will add to their respective sets. Each party also creates several more dummy values to add to its own set. The parties send all their hash values, dummy values and number 12, to the decider. The decider compares the hashes and observes that there are 15 hash values that are in both sets. Then the decider computes 15 - 12 = 3, and gets the real cardinality of S_T. Please note that without the use of dummies, the decider learns the cardinality of sets A and B and the hash values of the elements in the output set.

5.1.2 Private Membership Test

Another special variant of a PSO problem is the private membership test. We formalize the problem of PMT as follows.

In a PMT protocol one party (e.g., a server) has a database and the other party (e.g., a client) has only one item. Both the client and the server do not want to reveal their items. After executing a PMT protocol the client only learns whether their item is included in the server's database. The server does not learn the client's item nor whether it was in the database or not.

To motivate the problem of PMT, consider a scenario where a DNA sample has been found in a crime scene, and the local police wants to compare the sample with the Interpol's database of DNA samples. Neither party involved in this scenario wants to reveal their data to the other party. With a PMT protocol, the local police can query the Interpol's database in a privacy-preserving manner.

In Publication I, we proposed three PMT protocols with the emphasis on lowering the computation costs. We further continued our research on PMT protocols with the emphasis on reducing the communication complexity and our result is presented in Publication II. The work on PMT protocols is further developed in academia, for example, in [103, 104].

Encrypted Database

In the protocols of Publication I, the server encrypts the database in two ways: i) inserting the database in a Bloom filter and then encrypting the filter by Goldwasser-Micali Homomorphic Encryption (or alternatively with oblivious pseudorandom function), or ii) encrypting the database by using the RSA blind signature and then inserting the encrypted database into a Bloom filter. For both cases i and ii, the encrypting and building the Bloom filter is done during the off-line phase of the protocol. Then, the server delivers an encrypted version of the database to the client. In other words, the client receives the encrypted Bloom filter (or alternatively the Bloom filter that represents the encrypted database) and its hash functions.

By feeding the query item (or alternatively the encryption of the query item) to the hash functions, the client knows which indices in the filter are related to their query item. Therefore, in the on-line phase of the protocol the client only needs to decrypt the values of those indices with the help of the server. The client masks the indices, such that it is not possible for the server to guess them[3], and sends them to the server. In this stage of the protocol, the interaction between the client and the server results in the decryption of the desired indices.

After decrypting the corresponding indices, in the case they all have value one, the client learns that the item is likely in the database[4]. Otherwise, the item is not in the database.

Figure 5.1 shows a summary of the PMT protocols of Publication I. In these protocols, the most time consuming part of the computation is the encryption phase, and it is done only once in the set-up phase of the protocol. After that, it is possible to query the Bloom filter many times. However, by every query the client learns more about the filter. Arbitrarily many queries cannot be allowed because eventually the client would learn the entire Bloom filter. The communication complexity of these protocols is the same as the size of the Bloom filter.

[3]The details about how to mask the indices are presented in Publication I.

[4]As explained before, a query from a Bloom filter may result in a false positive.

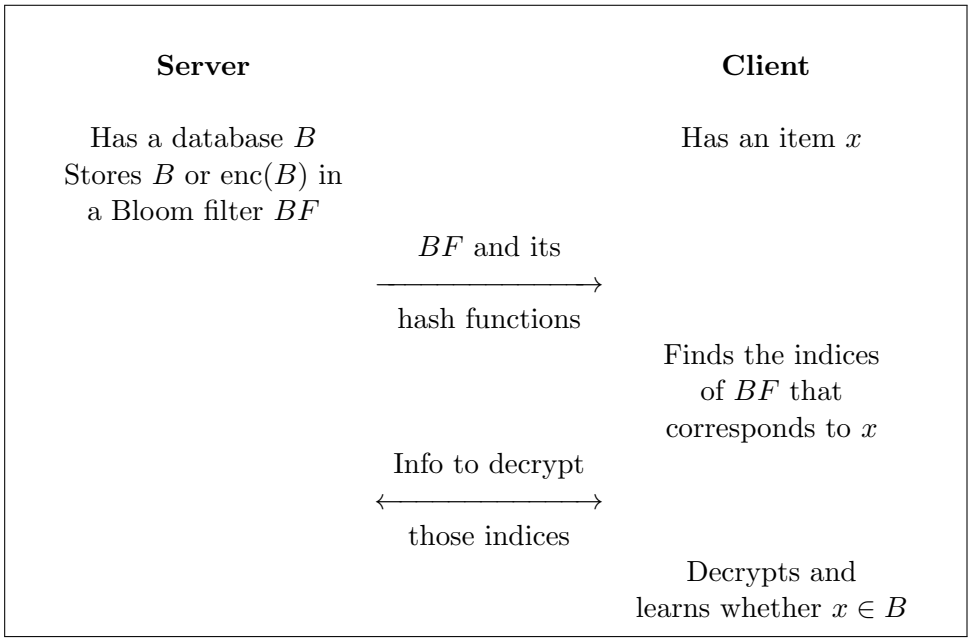


Figure 5.1: A summary of PMT protocols of Publication I.

Utilizing Homomorphic Encryption

Now, let us assume a setting in which the communication complexity of the protocol should be minimal. To design a PMT protocol with emphasis on lowering the communication costs, we use homomorphic encryption together with the Cuckoo/Bloom filter.

In the off-line phase, the server divides the database into 2^{2a} subsets such that each subset holds the elements of the database that all start with the same prefix of $2a$ bits. The server inserts each subset into a separate Bloom/Cuckoo filter. Please recall that each Bloom or Cuckoo filter is an array of several bits. Therefore, a Bloom/Cuckoo filter can be represented with a string of bits. Each bit string can be considered as a binary integer. Later in the protocol, we want to encrypt these integers that represent the filters. Therefore, based on the cryptosystem that is used in this protocol, the integers should be of a proper size so that they can be used as plaintexts. For instance, if the Paillier cryptosystem is used in our PMT protocol, the size of the integers should be less than the size of the Paillier private keys. If the size of these integers (and consequently the filters) are bigger than what is allowed by the cryptosystem, the server first divides the filters into b parts, such that each part has the proper size of a plaintext. Note that

the concatenations of all these b parts result in the original Bloom/Cuckoo filter.

The server $\mathcal{S}$ creates b matrices of size $2^a \times 2^a$ where the elements of these matrices are the integers corresponding to the parts of the filters that have been made in the previous step.

The client $\mathcal{C}$ has a value x and based on the value of the first $2a$ bits of x, knows which index of the matrix (let us say index (i^*, j^*)) corresponds to this value x. In other words, the first a bits of value x is equal to i^* and the second a bits of x is equal to j^*.

In the on-line phase of the protocol, the client retrieves the values of the matrix elements at position (i^*, j^*) in all the matrices, utilizing Chang's PIR protocol [66]. The client concatenates these values and obtains a Bloom/Cuckoo filter that represents the subset of the database that has all the elements with the same prefix of $2a$ bits, as x has. Now, the client can look for x in this filter. Figure 5.2 shows an overview of this protocol.

This protocol is extended such that it works with data structures that have $N > 2$ dimensions. We implemented the extended protocol and executed it for different dimensions and database sizes. We wanted to find out which dimension gives the best performance in our protocol. We observed that the optimal number of dimension depends on the size of the database and whether it is the time complexity or space complexity that is more important to minimize. The complete results of performance evaluations are presented in Publication II.

5.1.3 Criteria to Classify Private Set Operations

As we discussed before, Private Set Operations have many use-cases. Different use-cases in which a PSO protocol is useful, lead to different variants of PSO protocol. Now, we present some of the criteria that can be used to classify the PSO settings:

1. Identifying which of the following information is required about the output set: i) the whole set, ii) the cardinality of it, or iii) whether it is empty or not.

2. The party or parties who get the final result.

3. The adversarial model in which the protocol is run.

4. The size of the universe in which the elements of the private sets belong.

5. The number of parties that have private sets and are participating in the protocol.
6. The set operation that is required for each specific use-case.
7. The size of the input sets in comparison with each other.
8. The availability of a trusted third party.
9. Whether the parties who are going to participate in the protocol, or some of the input sets (if not all) are known in advance (in the off-line phase).

In order to give an example of the possible combinations of the above criteria, let us take another look at the motivational example that is given in the beginning of this section: a statistical institute of a region wants to learn the performance of the health care system in that area. In this example, the statistical institute learns the cardinality of output set (Criterion 1 and Criterion 2), and we assume that the parties follow the protocol honestly (Criterion 3). Hospitals and insurance companies have private sets, which are a subset of the citizens of that region, therefore, the size of the input sets are relatively similar, and the universe has a limited size (Criterion 4, Criterion 5, and Criterion 7). The required set operations are: unions of patients with certain diseases (set A), unions of people who are covered with an insurance (set B), and finally the intersection of sets A and B (Criterion 6). There is no trusted third party (Criterion 8), and the input sets are known in the off-line phase of the PSO protocol (Criterion 9).

In order to determine which PSO protocol works best for a specific application area, we first need to determine how each criterion can be applied to that application area.

5.2 Privacy-Preserving Graph Queries

The importance of data availability and utilizing graph structures to provide proper access to information motivate researchers to develop efficient query techniques on graph data structures. As one research theme, privacy-preserving protocols on graph data structures have been studied extensively. Some examples of settings in which a privacy-preserving graph query is needed are: graph intersection [105, 106], query on two graphs [107], spectral graph analysis [108], on-line social networks [109, 110], and web algorithms [111].

This section presents three privacy-preserving protocols that can be applied to any use-cases where the data can be presented using a graph.

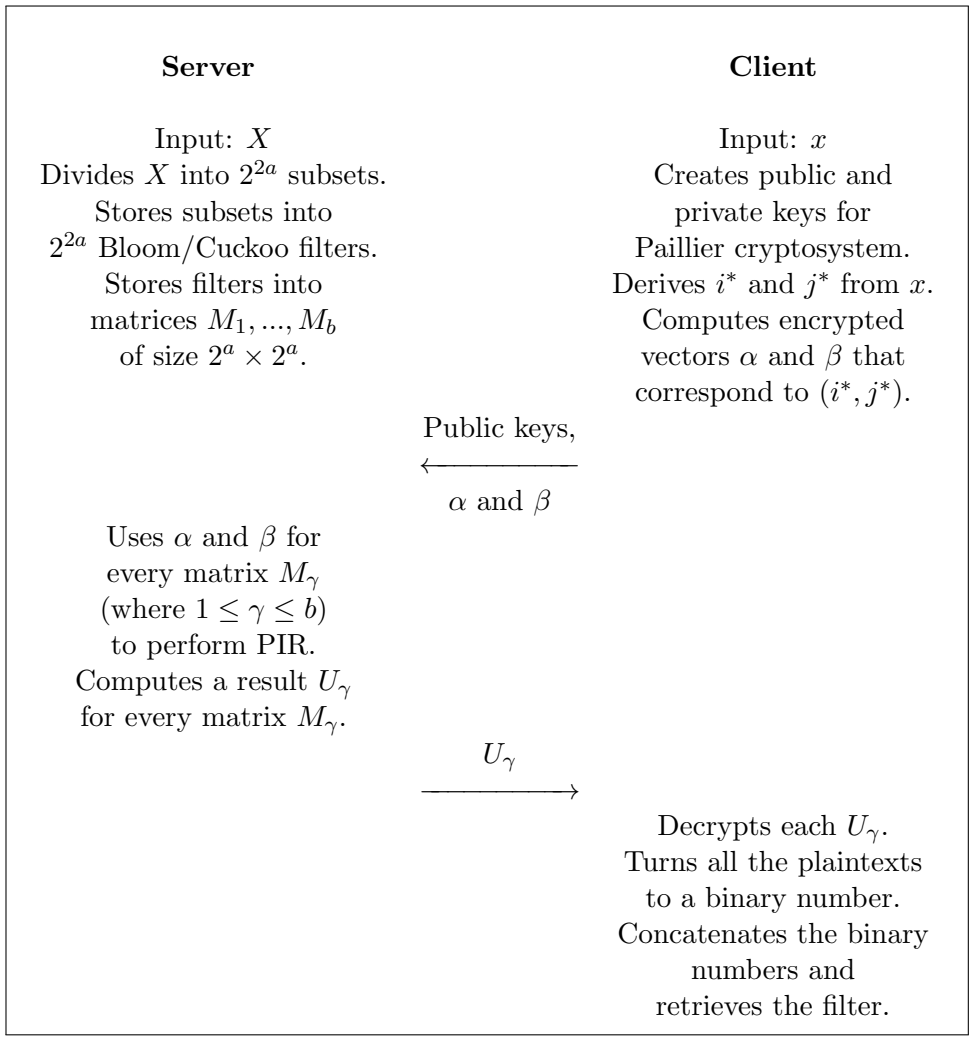


Figure 5.2: An overview of our PMT protocol on 2-dimensional data structures utilizing Paillier cryptosystem [61].

The results of our first protocol determines whether there is a path from node i to j. The second protocol retrieves the path from i to j, and the third one presents a 2-party query on a bipartite graph.

5.2.1 Existence of a Path

Let us assume a setting in which the database of sensitive data can be illustrated as a directed graph. In this setting, the database owner wants to utilize a cloud to store the database. Moreover, the owner wants to enable the cloud to help with the queries on the database. Lastly, let us assume that the queries on the graph are done only to determine whether there is a path from a node i to a node j. For this setting, we require a privacy-preserving protocol that enables queries on a directed graph in the presence of a cloud.

To motivate the problem of privacy-preserving path existence queries on graphs, let us consider the following scenario: In a certain geographical location there is a dangerous virus that is spreading via human contact. The virus can hide itself in a human body for a certain amount of time, e.g., for x days. Therefore, a person who carries the virus but looks healthy, can infect other people who are coming into contact with him/her. In order to monitor the situation, the local health organization decides that the people who have had any physical contact with the infected people, should be quarantined for the same amount of time that the virus hides itself in its host's body (x days). This data can be illustrated as a directed graph[5], where the nodes are all people in that area and the arcs are between two individuals who have had face-to-face contact during the past x days. We must consider in which order people have met each other. For instance, if j meets a healthy person k, and then meets the infected person i, then j has not infected k. Therefore, in this graph timestamps must be in chronological order, and directions should be included.

After constructing the above graph, in order to determine who has had contact with an infected person i, it is enough to find all the nodes that there is a path from i to them, while considering the time stamps[6]. As humans' social life is considered private, the health organization requires a privacy-preserving protocol to query such a graph to determine whether there is a path from infected i to another person.

[5]Please note that any undirected graph is directed when the edges are double-headed.

[6]Assume an infected person i met a healthy person f at 1 pm, the now-infected f met a healthy person g at 3 pm and still-healthy g met a healthy person h at 2 pm. In the graph, there is a path from i to h but the time stamps are not in "correct" order. If g met h again at 4 pm then there is an infecting path.

In a directed graph, if there is a path from node i to j and there is a path from node j to k then that means there is a path from node i to k. Therefore, we have transitivity and we can create a transitive closure graph from any (directed) graph where i and j are connected by an arc if there is a path from i to j. Moreover, we can create a matrix from the transitive closure graph of n nodes: We first create a matrix T of size $n \times n$. Then, $T_{ij} = 1$ if there is a path from i to j, and otherwise, $T_{ij} = 0$. The main diagonal of matrix T is set to 1.

In the off-line phase of our protocol, the owner inserts the database into a directed graph, then creates the transitive closure graph, and inserts this graph into the matrix T. The owner encrypts the matrix utilizing Protocol 1 of Publication I[7]. Then the owner stores the encrypted matrix into a cloud.

In the on-line phase of the protocol, the client wants to retrieve one bit of this matrix. The client uses Protocol 1 of Publication I, retrieves the encrypted bit and with the help of the owner, decrypts this bit. A summary of this protocol is shown in Figure 5.3.

This protocol is presented in Publication III. The research on privacy-preserving path queries has been further studied, for example, in the context of e-healthcare systems [112].

5.2.2 Retrieving a Path

Finding the shortest path[8] in a graph [113, 114] or in an encrypted graph [115] has been studied previously. In this subsection, we present a privacy-preserving shortest path retrieval protocol in the presence of a cloud. We motivate this problem in the context of a navigation application which finds the shortest way between two locations. A user of this application wants to find the shortest journey between their origin and destination, without disclosing these two locations to anybody.

Our protocol consists of 4 phases. The first phase of the protocol is the off-line phase and the required set-ups are done in this stage. The on-line phase of the protocol consists of Phases 2, 3 and 4.

In Phase 1, we first construct a matrix T, as explained in Subsection 5.2.1. We also require another matrix to help the client to retrieve the desired shortest path.

[7]Although Protocol 1 of Publication I, which is a PMT protocol with Bloom filters and Goldwasser-Micali cryptosystem, was designed originally to encrypt bits of a Bloom filter, it can also be used to encrypt bits of any data structure, such as a matrix.

[8]There can be multipe shortest pathes between two nodes in a graph, however, in this case finding one is sufficient.

The Floyd-Warshall algorithm [116] gives the length of the shortest path between two nodes. We extend this algorithm such that it outputs a matrix P, where each value P_{ij} of P is the penultimate node in the shortest path from node i to j. If there is no path from i to j then $P_{ij} = \perp$. Note that the main diagonal of the matrix P is 0. The extended Floyd-Warshall algorithm is shown in Figure 5.4.

The database owner chooses a symmetric encryption method, and generates a different key k_{ij} for each pair of nodes (i, j), where $i \neq j$. When $i = j$, the owner creates a dummy key of all zeros, denoted by $\bar{0}$. The owner inserts all the keys in a matrix $\mathcal{B}$.

The owner further creates a matrix $\mathcal{P}$, where $\mathcal{P}_{ij} = (P_{ij}, k_{iP_{ij}})$ when $i \neq j$ and $P_{ij} \neq \perp$. Moreover, $\mathcal{P}_{ij} = (0, \bar{0})$ when $i \neq j$ and $P_{ij} = \perp$, and the main diagonal of the matrix $\mathcal{P}$ is marked with ϵ.

Then the owner creates an encrypted matrix A, where the main diagonal of this matrix is marked with ϵ. The other entries of A are encryptions of entries of $\mathcal{P}$ by using keys of matrix $\mathcal{B}$, such that $A_{ij} = E_{k_{ij}}(\mathcal{P})$. Finally, the owner chooses public keys (e, n) for RSA encryption scheme with private key d, and encrypts the keys of matrix $\mathcal{B}$. The database owner inserts the encrypted keys into a matrix K. Finally, the owner sends the public key (e, n) and the encrypted matrices A and K to the cloud. The set-up phase of the protocol is now done. Figure 5.5 shows a summary of the set-up phase of our protocol, for a small directed graph that has only four nodes.

In the on-line phase of the protocol, a client who is interested in the shortest path from node i to j, needs to execute Phases 2-4. In Phase 2, the client receives the public key (e, n) from the cloud, and uses a PIR protocol on matrix K to retrieve K_{ij}. In Phase 3, with the help of the database owner and by using RSA blind decryption, the client decrypts K_{ij} and gets k_{ij}. In Phase 4, the client and the cloud execute a PIR protocol on matrix A such that the client retrieves A_{ij}. Figure 5.6 shows a summary of the 4 phases of our protocol.

The client uses the key k_{ij} from Phase 3 to decrypt A_{ij}, and therefore obtains $\mathcal{P}_{ij}$, and consequently the last node in the path from i to j before j and the key $k_{iP_{ij}}$. If $k_{iP_{ij}} = \bar{0}$ the protocol stops. Otherwise, the client repeats Phase 4 until the whole shortest path from node i to j has been obtained. This protocol is presented in Publication IV.

5.2.3 Retrieving a Path in a Bipartite Graph

Privacy-preserving protocols with bipartite graphs have been studied in different contexts, such as matching framework for multimedia analysis [117] and social media [118]. In this subsection, we present a general protocol to

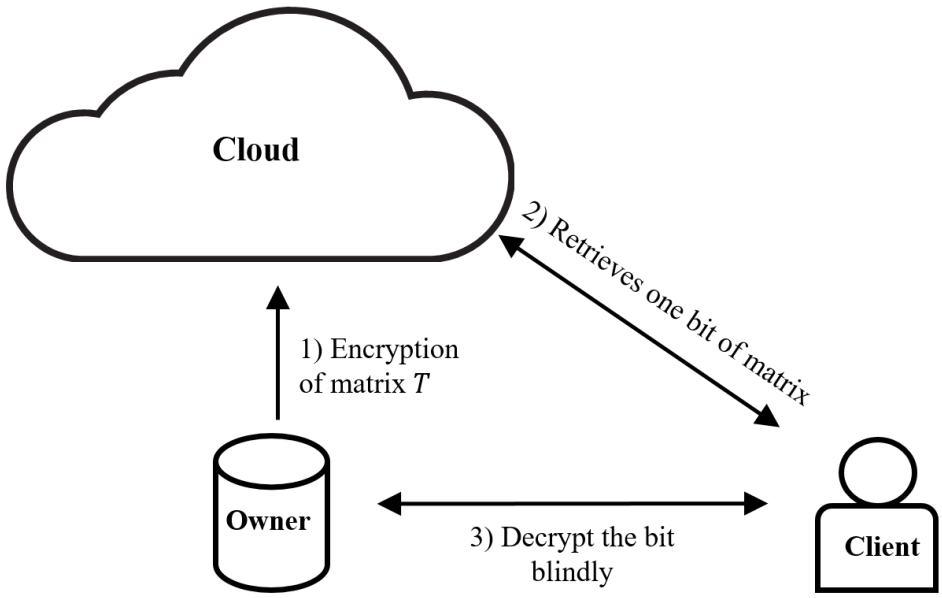
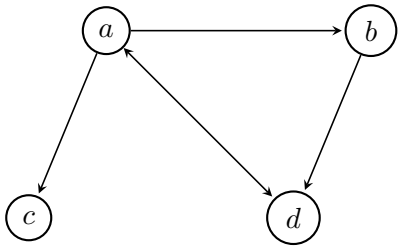


Figure 5.3: An overview of our privacy-preserving query protocol to determine the existence of a path.

```
for i from 1 to v:
for j from 1 to v:
if i == j:
W_ij = 0
P_ij = 0
else if (i, j) is an arrow in the graph G:
W_ij = 1
P_ij = i
else:
W_ij = ∞
P_ij = ⊥
for k from 1 to v:
for i from 1 to v:
for j from 1 to v:
if W_ij > W_ik + W_kj:
W_ij = W_ik + W_kj
P_ij = P_kj
return matrix P

```

Figure 5.4: Extended Floyd-Warshall algorithm. In this algorithm, the value of entry W_{ij} in matrix W is the length of the shortest path from i to j in the graph G.



(a) Graph G

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & a & a & a \\ d & 0 & a & b \\ \bot & \bot & 0 & \bot \\ d & \bot & a & 0 \end{pmatrix} \end{array}$$

(b) Matrix P

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} \bar{0} & k_{ab} & k_{ac} & k_{ad} \\ k_{ba} & \bar{0} & k_{bc} & k_{bd} \\ k_{ca} & k_{cb} & \bar{0} & k_{cd} \\ k_{da} & k_{db} & k_{dc} & \bar{0} \end{pmatrix} \end{array}$$

(c) Matrix $\mathcal{B}$

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} \epsilon & (a,\bar{0}) & (a,\bar{0}) & (a,\bar{0}) \\ (d,k_{bd}) & \epsilon & (a,k_{ba}) & (b,\bar{0}) \\ (0,\bar{0}) & (0,\bar{0}) & \epsilon & (0,\bar{0}) \\ (d,\bar{0}) & (0,\bar{0}) & (a,k_{da}) & \epsilon \end{pmatrix} \end{array}$$

(d) Matrix $\mathcal{P}$

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} \epsilon & E_{k_{ab}}(a,\bar{0}) & E_{k_{ac}}(a,\bar{0}) & E_{k_{ad}}(a,\bar{0}) \\ E_{k_{ba}}(d,k_{bd}) & \epsilon & E_{k_{bc}}(a,k_{ba}) & E_{k_{bd}}(b,\bar{0}) \\ E_{k_{ca}}(0,\bar{0}) & E_{k_{cb}}(0,\bar{0}) & \epsilon & E_{k_{cd}}(0,\bar{0}) \\ E_{k_{da}}(d,\bar{0}) & E_{k_{db}}(0,\bar{0}) & E_{k_{dc}}(a,k_{da}) & \epsilon \end{pmatrix} \end{array}$$

(e) Matrix A

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccccccc} a & & b & & c & & d & \end{array} \\ \begin{pmatrix} \bar{0}^e & \bmod n & k_{ab}^e & \bmod n & k_{ac}^e & \bmod n & k_{ad}^e & \bmod n \\ k_{ba}^e & \bmod n & \bar{0}^e & \bmod n & k_{bc}^e & \bmod n & k_{bd}^e & \bmod n \\ k_{ca}^e & \bmod n & k_{cb}^e & \bmod n & \bar{0}^e & \bmod n & k_{cd}^e & \bmod n \\ k_{da}^e & \bmod n & k_{db}^e & \bmod n & k_{dc}^e & \bmod n & \bar{0}^e & \bmod n \end{pmatrix} \end{array}$$

(f) Matrix K

Figure 5.5: In this figure, we show a graph with four nodes, a, b, c, and d. The matrices that are created in the off-line phase of our protocol to retrieve the nodes of a path in a directed graph are also shown.

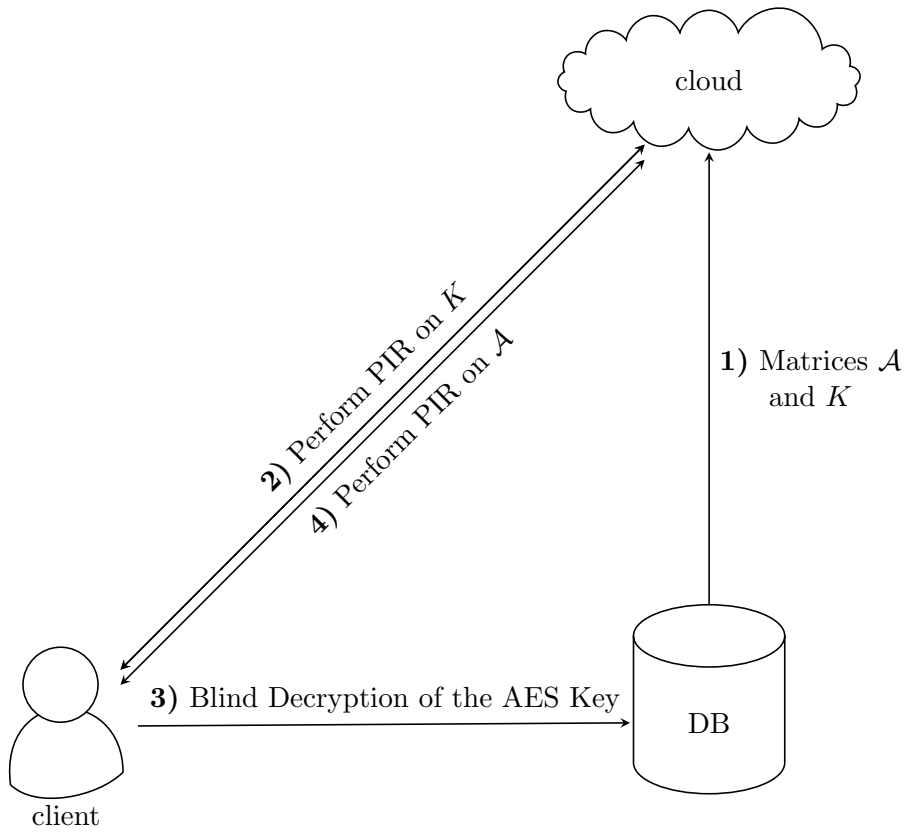


Figure 5.6: An overview of the 4 phases of our protocol to retrieve a shortest path in a graph.

perform privacy-preserving path queries on bipartite graphs. This protocol is presented in Publication V.

Let us assume that we have a directed bipartite graph G that consists of two parts U and V. The arrows of the graph G have their heads in U and tails in V or vice versa. The set U has two owners: A and B. The set U is divided into two subsets U_A and U_B such that $U_A \cap U_B = \emptyset$ and $U_A \cup U_B = U$. The set U_A belongs to owner A and the set U_B belongs to B. We denote the set of arrows in the graph G with E. Every arrow has one end in V and the other end either in U_A or in U_B. The set of former (latter, respectively) type of arrows is denoted as E_A (E_B, respectively). The set E has two owners: the set E_A belongs to owner A and the set E_B belongs to B. Moreover, $E_A \cap E_B = \emptyset$ and $E_A \cup E_B = E$. The nodes of the set V belong to both owners A and B. An example of a directed bipartite graph G is shown in Figure 5.7.

Owners A and B want to know whether there is a path from node $a \in U_A$ to node $b \in U_B$, in a privacy-preserving manner. In other words, the owners want to know if $a \in U_A$ can reach $b \in U_B$, without revealing their respective part of the graphs to each other.

In addition to the applications mentioned in the beginning of the section, we further motivate the study of privacy-preserving queries on bipartite graph, by looking at the motivational example of Subsection 5.2.1. In a pandemic situation, controlling the social contacts of one area is not enough. Assume two different countries want to privately determine which one of their citizens have come into contact with the other country's infected people. The graph of social contacts can be assumed as a bipartite graph that is owned by two owners; each owner is a healthcare system of one of the countries. The set U is all the infected people in both countries, and the set V is the healthy people of these countries.

In order to determine whether there is a path from node $a \in U_A$ to node $b \in U_B$, we first divide the possible routes from a to node b into two categories: a *single-crossing path* and a *general path*. We say there is a single-crossing path from a to b if there exists a node $v \in V$ such that by only using arrows in E_A we can reach from a to v. Then, we can reach from v to b by only utilizing arrows in E_B.

A general path is any kind of path from node $a \in U_A$ to node $b \in U_B$. In other words, a general path can be a single-crossing path, or can include multiple paths from E_A to E_B or vice versa. For an example of such multiple crossing path see the path from node $a \in U_A$ to node $b \in U_B$ in Figure 5.7.

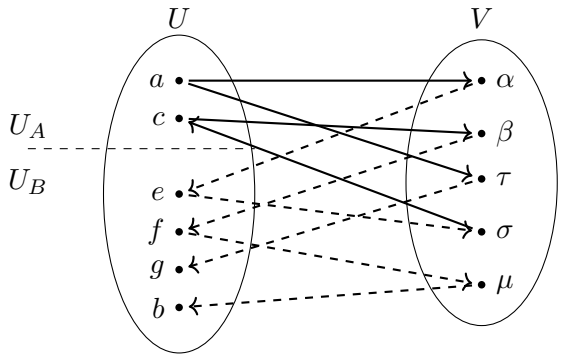


Figure 5.7: An example of a directed bipartite graph that has two parts U and V. Solid arcs belong to the set E_A and dashed ones to the set E_B. One path from node $a \in U_A$ to node $b \in U_B$ is $a\alpha e\sigma c\beta f\mu b$.

In Publication V, we presented two protocols to perform privacy preserving queries on bipartite graphs to know whether there is a path from node $a \in U_A$ to node $b \in U_B$. The first protocol determines whether there is a single-crossing path between these two nodes. In the first protocol, owner A finds out what nodes can be reached from a in A's part of the graph and owner B finds out what nodes reach b in B's part of the graph. To do so, owner A generates a set V_a which contains nodes $v \in V$ such that there is a path from a to v by utilizing only arrows in E_A, and owner B creates a set V_b which contains nodes $v \in V$ such that there is a path from v to b by only using arrows in E_B. Both parties together use a PSI between these two sets V_a and V_b. If there is a non-empty intersection between V_a and V_b then there is a single-crossing path from a to b. Otherwise, there still might be a path from a to b, but we require the second protocol to determine that.

The second protocol of Publication V determines whether there is a general path from node $a \in U_A$ to node $b \in U_B$. The idea behind this general protocol is similar to the single-crossing path protocol; the owners consider what nodes are reachable in their part of the graph and then together find out if there is a path from a to b. The general path protocol is done as follows.

By utilizing a PSI protocol twice, owners learn two sets of nodes: V_{AB} and V_{BA}. The set V_{AB} (V_{BA}, respectively) contains nodes $v \in V$ such that there exists an arrow $(u, v) \in E_A$ ($(u, v) \in E_B$, respectively) and another arrow $(v, f) \in E_B$ ($(v, f) \in E_A$, respectively). The nodes u are in the set U_A, and the nodes f are in the set U_B.

Now, with the help of the Floyd-Warshall algorithm and by utilizing only the arrows in E_A, owner A finds out which nodes of V_{AB} are reachable from a. We denote this set of nodes by V_0^A. If there is no such node, then there is no path from a to b, so owner A stops the protocol and informs B that there is no path. If $V_0^A \neq \emptyset$, then A sends V_0^A to B.

Owner B first checks to see if there is a path from a node in V_0^A to b . If there is a path, B informs the output to A and the protocol stops. Otherwise, B finds out which nodes of V_{BA} are reachable from V_0^A by only using arrows of E_B, and denotes this set of nodes by V_0^B. Owner B sends V_0^B to A. If V_0^B is empty, there is no path from a to b. Owner B informs A that there is no path, and the protocol stops. Otherwise, the owners continue the protocol for $i = 1, 2, 3, ...$ as follows:

1. A first looks for a path from a to a node in V_{i-1}^B. If such a path exists, A provides the path to B and the protocol stops. Otherwise, with the help of the Floyd-Warshall algorithm, owner A uses only arrows of E_A to find the nodes v that are in $V_{AB} \setminus \cup_{j=0}^{i-1} (V_j^A \cup V_j^B)$, and there is a path from some nodes in V_{i-1}^B to v. Owner A denotes this set as V_i^A. If V_i^A is empty, A stops the protocol and informs B that there is no path. Otherwise, A sends V_i^A to B.

2. Owner B tries to find a path from a node in V_{i-1}^A to b. If a path exists, B provides the path to A, and the protocol stops. Otherwise, using the Floyd-Warshall algorithm, B utilizes only arrows of E_B to find a set V_i^B that contains nodes $v \in V_{BA} \setminus \cup_{j=0}^{i-1} (V_j^A \cup V_j^B)$, that there is a path from some nodes in V_{i-1}^A to v. If V_i^B is non-empty the protocol continues, and B sends V_i^B to A. Otherwise, B stops the protocol and informs A that there is no path.

Please note that the above steps continue until a path from a to b is found, or a set V_i^A or V_i^B is found empty, which means there is no such path.

Compared with finding the general path, finding the single-crossing path is more privacy friendly, because the owners only learn whether the desired path exists or not. In the protocol to find the general path, the owner learns some information about the nodes that can or cannot be reached by the other owner. On one hand, from the privacy point of view, it is better to first execute the protocol to find a single-crossing path. If no such path has been found, then the owners can enter the second protocol to find out whether a general path exists. On the other hand, if the use-case of the protocol allows revealing some information about the parts of the graph to the other owner, the protocol to find the general path can help in finding any path that exists between the desired nodes.

Chapter 6

Application Areas

Throughout this dissertation, we discussed the importance of privacy in different settings several times. In Chapter 5, we gave more detailed examples to motivate our theoretical developments. The examples that we gave presented the importance of privacy in some application areas related to health care, forensic science, and route-finding systems. The findings of Chapter 5 have a much wider application area than we explained, for example, in the context of privacy-preserving networking via social media [90, 118], managing human resources in a private manner [119], and anonymous voting [89]. Going through all the application areas where our protocols are applicable is outside the scope of this dissertation.

Instead, we take a closer look at three particular application areas in which privacy enhancing technologies are needed:

- Malware Checking
- Protection of Remote Access
- Protecting Minors

As explained in Chapter 2, there are several benefits in utilizing distributed computing. Thus, there has been a trend toward utilizing cloud-based and edge computing services to store and query data. If the distributed computing helps, we want to design our solutions for each of the above application areas, such that they can benefit from advantages of distributed computing.

Privacy in distributed computing has been a subject of concerns [120, 121, 122]. Therefore, in the application areas in which distributed computing is used, the privacy of users and also data owners should be discussed.

In Chapters 1 and 4, we briefly explained the three application areas. In this chapter, we take a closer look at each of these areas.

In each application area we first give a more in-depth introduction and explain why privacy is important. Then we identify the challenges that we may encounter when designing a privacy-preserving protocol for that specific application area. Next, again for each application area, we present our protocols that can solve these real-life problems and we explain why our proposed solutions are better than their predecessors. All our solutions have been published in the publications that are part of this dissertation. Furthermore, we give the current state of the art of privacy in each application area.

6.1 Privacy-Preserving Technologies for Malware Protection

Malware refers to software that are produced with the intention to perform malicious acts on the devices that are infected by them [123]. A computer virus, spyware, and trojans are examples of malware [124].

A piece of malware can cause a variety of damages, and therefore it is important to design methods to prevent a system from being infected by malware, while also helping the infected systems to clean-up. However, just like preventing a human infection (and sometimes even a pandemic) is not always possible, preventing malware infections completely is not a realistic goal. Therefore, a regular check to detect possible malware in a system is necessary.

As mentioned before, there is a tendency towards utilizing cloud computing. This is the case also for malware protection. With the increase of cloud-based services, the demand to design and deliver fast and secure query techniques is also increasing. However, using a cloud has several challenges regarding security [125, 126]. There is also an increase in concerns about the privacy of the data owners and their customers who use cloud-based services [127, 128]. Customers want to have the option of making queries to the database without revealing their queries to the database owners or to the cloud. In this section we present several on-line privacy-preserving malware checking systems that utilize cloud computing.

6.1.1 Background on Malware Protection

With increasing use of internet in everyday life, attackers have access to an even bigger venue to spread malware. There are several methods to

detect malware, to protect digital devices from being infected by malware [79, 129], and to classify malware programs [130, 131]. The primary defense against malware infection is a malware detector.

Traditionally, an anti-malware program is stored in a client device. The anti-malware program has a local database of known malware samples and detects the presence of malicious software in a digital content (an application, a file, etc.). The client who has a digital content that is marked malicious by the local anti-malware program can do the following. The client who wants to be certain whether the content is indeed malicious, would deliver the content to an anti-malware server. The client considers the anti-malware server to be a trusted party. The server then determines whether there are any traces of malicious software in the client's content. Finally, the system informs the client whether the content is clean or not, and helps the client to clean-up their content (and if necessary also the whole device). Therefore, in this process the user privacy is not protected towards the server.

6.1.2 Privacy in Malware Protection

In order to explain the importance of privacy in the context of malware checking, let us consider a scenario where a client $\mathcal{C}$ wants to install an application on their smartphone. A server $\mathcal{S}$ has a database of malicious applications. The client wants to check whether the desired application is malicious, by looking for it against the server's malware database. While the client $\mathcal{C}$ queries the server's database, $\mathcal{S}$ learns which application $\mathcal{C}$ is interested in, and therefore, might learn about this client's religion, location, relation status, political views, etc. [12]. This scenario shows the importance of a privacy-preserving method to enable malware checking queries.

Private membership tests are applicable in many real-life applications, e.g., in the context of designing query-response protocols for cloud-assisted services. In this section, we discuss malware checking in a privacy-preserving manner by utilizing PMT protocols.

Please note that the presented protocols are not limited to malware checking services and can be used for any scenarios that require a PMT protocol. The following examples are a few application areas where a privacy-preserving membership check can be used: checking the name of a "person of interest", checking a DNA sample or a fingerprint against Interpol's database of criminals, checking a password against a list of leaked passwords.

6.1.3 The Protocols

In this dissertation we only want to check a digital content for possible infection by known malware. Therefore, the setting of the privacy-preserving malware checking problem is similar to the setting of a PMT problem: In both cases, one party (in the case of privacy preserving malware checking, the server) holds a set of private elements (known malware samples) and the other party (the client) wants to privately look for a specific item (an application). On one hand, in the malware checking scenario false negative results are not tolerated. On the other hand, a small percentage of false positives is tolerable[1]. This means that the server can use a Bloom or a Cuckoo filter to store the malware samples. Therefore, the PMT protocols of Subsection 5.1.2 can be used for the purpose of malware checking. Please note that for security reasons (such as not to spread the malware samples accidentally), the server stores the fingerprints of the malware samples in its database.

By using one of the methods explained in Chapter 5, the server can encrypt the database of malware samples in one of the following ways: i) the server uses a Bloom filter to store the malware samples and then encrypts the filter or ii) the server first encrypts the database and then stores the encrypted database in a Bloom filter. Then, the Bloom filter is stored in a cloud. The filter only contains encrypted values and therefore, it can not reveal the malware samples to anybody who may access this filter.

A client $\mathcal{C}$ wants to see whether an application is clean or not. $\mathcal{C}$ downloads the Bloom filter from the cloud and engages in a series of interactions with $\mathcal{S}$ to decrypt the required bits of the filter. This approach is presented in Publication I.

The server can also provide privacy-preserving malware lookup services by utilizing the homomorphic encryption method of Chapter 5. The server divides the malware samples into several subsets, and stores each subset in a Bloom or Cuckoo filter. Then, $\mathcal{S}$ inserts the filters into a matrix. The matrix is stored in a cloud. The client uses Paillier homomorphic encryption and retrieves one entry of the matrix. The client receives the hash functions related to the filters from the server and is able to look for the fingerprint of x. Moreover, we generalized the homomorphic encryption approach to N-dimensional databases. This solution is presented in Publication II.

[1] If the results of a malware checking is positive, then the client can ask the server for further analyse of the application. After checking the application by the server, if it turns out that the result was a false positive, then the server informs the client that the application is clean. That's why a small percentage of false positives is tolerable in our protocol.

Privacy-preserving malware checking with database encryption is suitable for the use-cases where the set of known malware is not changing too frequently. This is due to the fact that when there are too many updates in the original database, the Bloom filter which represents that database (or the encrypted database), is changed often as well. This means the client might have to download a new filter upon each query.

On the other hand, the advantage of the encrypted database approach of Publication I compared to using the homomorphic encryption approach of Publication II, is in the amount of information that is revealed each time that the protocol is executed. With the homomorphic encryption method, the client retrieves a smaller Bloom/Cuckoo filter that represents a subset of the database that corresponds to the client's desired item[2]. The client can investigate further into the filter that is retrieved, and look for other items that correspond to the filter. However, with the encrypted database method, the client only learns whether or not their item is in the server's database. Therefore, the encrypted database method reveals only the results of the PMT and preserves the privacy of the server more that the homomorphic encryption approach. However, the homomorphic encryption approach has significantly smaller communication complexity than the encrypted database method. In conclusion, one of the above-mentioned methods can have advantages over the other one depending on the use-case and the server's privacy requirements.

6.2 Privacy-Preserving Technologies for Protection of Remote Access

In this section we present methods to perform privacy-preserving queries on a database of trust relations.

In Chapter 4 we defined a trust relation as a set of rules that grant permission to a user at a host to remotely access another user at a different host via an authentication key that has a unique fingerprint f. We choose a quintuple of form (source-user, source-host, fingerprint, target-user, target-host) to represent a trust relation. A database that consists of trust relations is a set of quintuples. Each quintuple in fact consists of a pair of triples: (source-user, source-host, fingerprint f) and (fingerprint f, target-user, target-host). A database of trust relations can be seen as a directed graph where each user-host pair is a node in the graph and the fingerprints are the edges that connect these nodes.

[2]Please recall from Section 5.1 that the filter that is retrieved by the client contains all items that have the same prefix of $2a$ bits that the client's desired item has.

With the widespread utilization of the internet, every day there are more devices that connect to a certain domain. Hence, the size of the trust relation database is increasing in average. For secure networking, it is crucial to constantly remove the trust relations that are not needed any more. All together, it is important to develop techniques that support trust relation database management.

Please note that the importance of techniques discussed in this section are not limited to trust relations. Any data that is privacy sensitive, and can be illustrated as a graph can benefit from these techniques.

6.2.1 Background

Generally speaking, query optimizations [132] and database management [133] are among the challenges that are faced when designing techniques for retrieving information from a database. The database model that is chosen to represent a database has an impact on the managing of that database [134]. We choose graph database models to illustrate the trust relations.

Graph database models have been used widely to represent a variety of databases where the linkages between data are as important as (or even more important than) the data itself [135]. Although, to the best of our knowledge, there are no studies on privacy preserving queries on remote access rules, the variety of literature on graph database optimization, queries and management can be applied to the topic of the present section of this work, such as [136], [137] and [138].

6.2.2 Privacy in Protection of Remote Access

In this subsection we explain why designing privacy-preserving protocols to query a database of trust relations is necessary. Let us give an example to explain the importance of trust relational look-ups. Assume there are two sets of trust relation databases S_1 and S_2, each with its own administrator; A_1 manages S_1 and A_2 manages S_2. Please recall that a database of trust relations consists of quintuples of the form (source-user, source-host, fingerprint, target-user, target-host). Assume a user U is a source-user/target-user in both databases S_1 and S_2. User U's access rights in S_2 are expired. After the expiration is detected, A_2 terminates U's access right and blocks the access from U to other users of S_2. Admin A_2 can block U's access to users of S_2 by removing some quintuples in S_2. However, since U's access rights are still valid in S_1, user U might be able to access other users of S_2, via some users of S_1. For example, assume U can access $a \in S_1$ and a can access $b \in S_2$, then U can access b. Admin A_2 does not want to

reveal to A_1 which user's access rights have been terminated. However, A_2 wants to look out for the possible indirect access from U to its users via S_1. Therefore, the admins require a privacy-preserving query technique to perform their look-up.

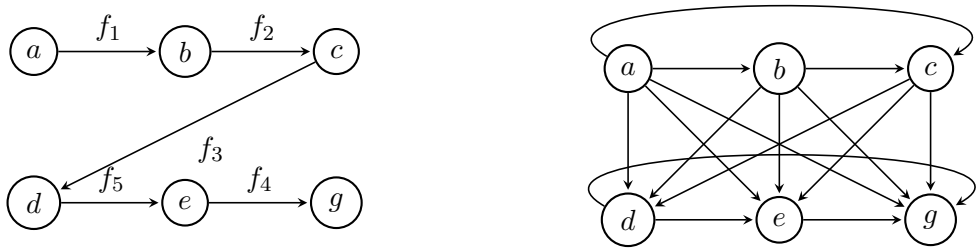
Let us look at another example where someone else than the admin of a trust relations database wants to make a query. Assume there is an admin A_1 that manages trust relations database S_1. Assume there is a certain target-user V in S_1 which is considered to be a special user and only a handful of users are authorized to access V . Now, an entity (e.g., police) wants to make sure that there is no unauthorized access to V, without revealing to the admin which user is the special target-user. In this scenario, the entity requires performing a privacy-preserving path query from any unauthorized source-user in S_1 to V to determine whether unauthorized access is possible.

6.2.3 The Protocols

In this subsection we formulate the problem of privacy-preserving queries on a trust relational database. As we mentioned, the owner of the database can illustrate the database of trust relations with a directed graph G, such that each pair of (user, host) is a node in this graph, and arrows are drawn where there is a trust between two (user, host) pairs. The arrows can be labelled with fingerprints.

In order to determine whether user A at a host has permission to access a user B at another host, we can look for a path from node A to B in the directed graph G. The graph of trust relations can contain millions of nodes and arrows, and therefore, we want to insert it into a more manageable data structure. One possible way to store the graph G is explained in Subsection 5.2.1: We can make a transitive closure graph from the graph of trust relations, as it is shown in Figure 6.1, then we can insert the transitive closure graph into a matrix, Now, the protocol of Subsection 5.2.1 can be used to privately retrieve one bit of this matrix with the help of a cloud, and consequently to confirm whether there is a path from A to B. Publication III is related to this application area.

Now let us assume a case where there is a trust relation from user A to user B, and from user B to user C. In other words, there is a path from A to C in the transitive closure graph that is obtained from G. An administrator of the trust-relational database may want to learn all the intermediary nodes (users) in a path from A to C. Subsection 5.2.2 presented a protocol that uses a cloud to privately retrieve all the nodes in a path from A to C. The protocol of Subsection 5.2.2 is also presented in Publication IV.



(a) Graph G of trust relations

(b) Transitive Closure Graph of G

Figure 6.1: Design of a Transitive Closure Graph from the Graph of Trust Relations.

A trust-relational database can have two (or even more) owners. Now, we assume that the database has two owners A to B and each owner has a set of users that only belongs to that owner. Therefore the graph that is made from this database is a bipartite graph that has two parts U and F. Set U is the set of users and F is the set of fingerprints. Owner A has a set of users U_A and B has a set of users U_B. The intersection of U_A and U_B is empty and their union is U. The owners want to perform 2-party queries on their bipartite graph in a privacy-preserving way. In Subsection 5.2.3 we presented a protocol that fits into this setting. This application area is detailed in Publication V.

6.3 Privacy-Preserving Technologies for Protecting Minors

As mentioned before, the rise of digitalization has both advantages and challenges. One of these challenges is how to protect the users of digital technologies. Children (minors) are a type of users in the digital world that are at risk the most.

Firstly, the 5G networks promise faster and more reliable services, and secondly, every day, there are more and more digital technologies that are developed. Thus, our everyday lives are now even more dependent on the internet. Therefore, children spend more time on-line than ever before [139, 140], and consequently, techniques that can guarantee a minor's protected networking and preserve their on-line privacy is now more important than ever. In this section, we present methods to keep children safe in the on-line world, in a privacy-preserving way.

6.3.1 Background

The digital *parental control* techniques [141] refer to technologies that protect children in the on-line world. Studies show that children who have been exposed to harmful on-line content, such as violence and adult content, can possibly suffer from its negative psychological effects for a lifetime [142, 143, 144]. Therefore, it is important to prevent the exposure of children to potentially harmful digital materials.

In 2015, Fuertes et al. [145], presented a comprehensive study on the accuracy, security and functionality of some of the well-known parental control tools. They also conducted several surveys and showed that many parents do not use any parental control tools, and do not know how to block harmful digital content. Their study shows the importance of an automatic parental control tool that is available to all children, without the need for parental intervention.

There are several application-specific studies of parental control, for instance, in the context of YouTube Kids [146] and smart toys [147]. These studies show the advancement of digital media, while highlighting the importance of parental control.

The effect of parental control is also studied by psychologists. Some of these scholars cite the negative impact of parental control on the child's privacy [148, 149]. However, to the best of our knowledge, there are no studies of privacy-preserving parental control methods.

6.3.2 Privacy in Protecting Minors

There have been several studies on the shortcomings of the current digital parental control techniques [150, 151, 152]. From our point of view, there are three main problems with the current parental control techniques:

- First, most of the parental control applications give a comprehensive report of the child's on-line activities to the parent. For instance, if a child makes a phone call, the parental control application records this call and sends a copy of that to the parent. Therefore, the child who is under parental control, has no privacy.

- Second, these techniques are only available to children whose parents can and want to utilize parental control techniques to influence their child's on-line life experiences. Not all parents can afford the extra expenses that are related to these techniques, and not all children are being raised by their parents. Therefore, not all children can benefit from the current parental control services.

- Third, many parental control services do not act in real time and only inform the parent about the incident after the child has been exposed to the harmful content. For instance, most of the parental control applications only deliver a copy of the text messages that the child sent/received to the parent. If any of those text messages contain harmful content, the child has already been exposed to the potential damage before the parent is being informed.

In the following subsections, we present protocols for secure networking by protecting minors automatically in a privacy-preserving manner. We first discuss automatic protection against cyberbullying. Then, we present automatic parental control tools with 5G networks for protected web-surfing and protected text messaging.

6.3.3 Protocol for Protection against Cyberbullying

Traditionally, bullying was only limited to the times that an individual was socializing in the real world. However, on-line social networking creates bigger and more easily accessible platform for bullies to harass their victims [153]. On-line bullying or cyberbullying can happen every hour of every day, and it is possible for the bully to hide his/her identity in the internet.

Bullies now have a bigger venue to access their potential victims [154]. It has been found that victims of bullying and bullies themselves can suffer a long-term negative impact on their health and social interactions [155]. Therefore, in addition to providing educational means for parents and children to help to prevent bullying in school and during childhood [156], it is a good idea to block potentially bullying contents before they reach their targets.

Nowadays, the AI-based language processing tools make it possible to detect bullying in real time and automatically. On the other hand, as explained in Chapter 2, the softwarization of 5G networks makes it possible to add new functions to the network, easier than the previous generations of mobile networks. In other words, it is possible to design an AI-based cyberbullying function, and introduce it in 5G networks such that the automatic cyberbullying protection is available for all children.

In this subsection, we present a protocol to automatically prevent cyberbullying in 5G networks when the users are sending or receiving text messages. As personal text messages are considered to be private, we want to enable privacy-preserving cyberbullying detection. This protocol is presented in Publication VI.

In our cyberbullying prevention protocol we assume that the operator is able to give a cyberbullying-related label to each subscriber. The possible labels are *new, normal, victim,* and *bully*. When the new cyberbullying function is introduced to the network, all the subscribers are labelled *new*. The output of our protocol will show the operator whether a text message is potentially harmful. If a message is categorized as harmful, the message will be blocked. After a certain amount of communications, the subscribers that are often sent harmful messages, will be labelled *bully* or *victim* according to the results of our protocol, and the messages concerning these two categories of subscribers will always be checked. The subscribers that never send bullying messages will be categorized as *normal*, and their messages will be checked randomly. Please note that the labels will be checked frequently and if needed the operator updates the labels. Whenever a new subscriber joins the network, it will be labelled *new*. The operator executes our protocol for all the new subscribers to determines which of the labels *normal, victim,* or *bully* is suitable for each of the new subscribers.

In order to check a message for harmful content in a privacy-preserving way, we create a new component in the architecture of 5G that is called *Filter Check* (FC). The new component FC communicates with UPF and has the functionality of collaborating with UPF to perform a PSI protocol between the message (that is held by UPF) and a set of known cyberbullying keywords (that is held by FC).

As mentioned before, we want to use AI methods[3] to detect the presence of bullying in a message automatically. Therefore, we create another component that is called *cyberbullying prevention function* (CBPF), that communicates with the functions of core network and performs the language processing of the message.

Our protocol has four phases; Phase 0 is the off-line phase of the protocol and Phases 1-3 are the on-line phases:

- **Phase 0:** In this phase, FC performs the necessary set-up operations for the PSI protocol, and UPF learns the label of the subscribers and therefore knows whether or not to perform Phase 1.

- **Phase 1:** If UPF decides to perform a cyberbullying check, then it enters Phase 1, where the PSI protocol is performed. Utilizing PSI to detect the traces of bullying in a message makes our protocol privacy friendly. If the result of the PSI protocol is an empty set, then the message is forwarded to its receiver. Otherwise, UPF enters Phase 2.

[3]Please note that in this dissertation we do not detail the AI part of the protocol. For more information about the AI part please see Publication VI.

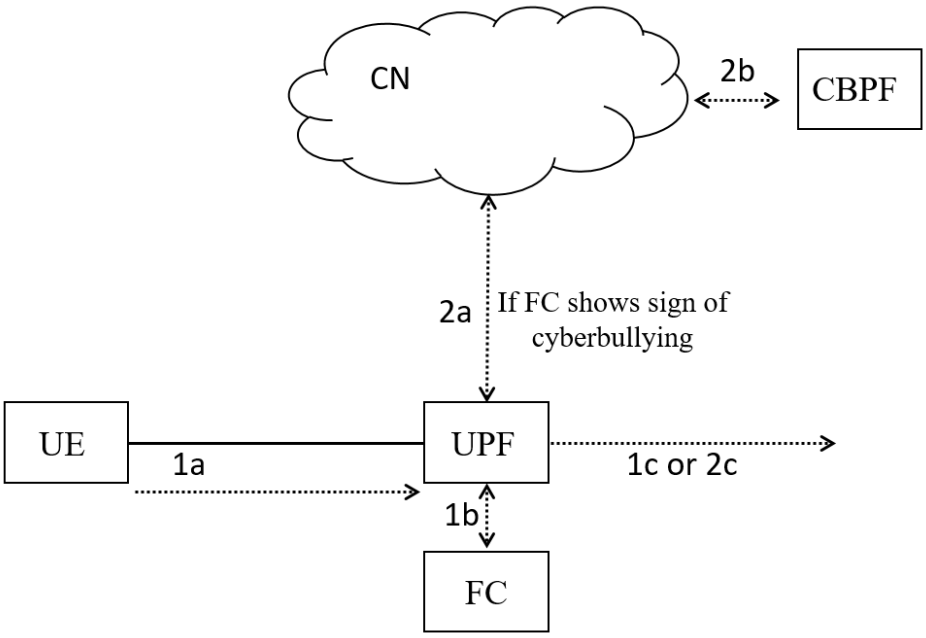


Figure 6.2: An overview of Phases 1 and 2 of our cyberbullying prevention protocol.

Please note that by performing Phase 1, we want to guarantee privacy for messages that are not detected to be harmful. If the result of PSI raises a red flag we further analyse the message to make sure whether or not it is harmful. Therefore, after Phase 1, the message is revealed to the operator for further analysing.

- **Phase 2:** UPF sends the message to the core network, where the message is classified as either bullying or benign by the CBPF component. If the message is benign, UPF forwards the message and otherwise it blocks the message.

- **Phase 3:** In the last phase of the protocol, based on the results of Phase 2 the operator decides whether or not to update the labels of the sender and receiver of the message. The detailed explanations regarding the updating of the labels can be found in Publication VI.

Although we use our protocol to protect minors from cyberbullying, the protocol can be used to detect and prevent cyberbullying for adults, as well. Figure 6.2 shows a summary of Phases 1 and 2 of our protocol.

6.3.4 Parental Control Protocol

In this subsection, we present two parental control protocols: the first protocol is used for protected web-surfing and the second one is utilized for protected text messaging. Each parental control protocol has two types. The first protocol type is called *General Check* and it is used in situations where the parents do not have any roles in their child's on-line activities. The second protocol type is called *Personalized Check*, and it is utilized for the cases where the parents want to have personalized influence on their child's digital life.

In our protocols we assume that each device operated by a minor includes an application that is called *Kid-client*. All the minor's on-line activities have to go through the kid-client. We also assume that each parent who wants to be involved in their child's activities on the internet has installed an application that is called *Parent-client*. The parent stores their wishes regarding their child's digital life in their local parent-client application.

We use edge computing in our protocols. We assume that the data related to each child is handled by an edge server. In order to make the parental control protocols detect harmful contents in real-time and automatically, we use AI techniques. We assume that these AI techniques are performed at the edge of the network, close to where the child's device is located. Moreover, we assume each edge server has a parental control function that participates in the parental control protocols. The operator assigns an edge server to each kid-client. The edge server is chosen based on the child's location, i.e., the edge server which is closest to the child's device handles the traffic that is generated by the child's device.

In our protocols, instead of blocking harmful content we use a *smart response*. The smart response is a positive digital content that replaces the child's original request, e.g., the content of a kid-friendly website is shown instead of a potentially violent website[4].

Our parental control protocols are detailed in Publication VII. Please note that these protocols can also be used for other use-cases than parental control, for example, to detect spreading hate speech and fake news via web pages and text messages. The importance of a centralized parental control system is realized in the 3GPP documentations [157].

[4]For more information about the smart response please see Publication VII.

Parental Control via Protected Web-surfing

The concept of categorizing web pages based on their content [158] has been researched extensively for a variety of purposes such as detecting unlawful on-line sales and detecting adult content.

In our protocol, when we classify web pages, we also consider the user's age. For instance, the content of a web page might be suitable for a 12-year old child but the same content is considered to be harmful for a 7-year old child.

Our general check protocol for the purpose of protected web-surfing is as follows: In the off-line phase of the protocol, with the help of AI, the operator classifies some popular or well-known web pages to *allow* and *deny* lists, based on their contents. These lists are created for every age group. The operator also creates AI-based smart responses for each age group. Finally, the allow and deny lists and the smart responses are sent to the edge servers.

The on-line phase of the protocol starts when the child wants to access a website. The web browser on the child's device sends the URL of the website to the kid-client, and the kid-client redirects it to the edge server that is assigned to this kid-client. The edge checks the URL against the allow list (deny list, respectively). If the URL is in the allow list, the edge sends the contents of the web page to the kid-client. If the URL is in the deny list, the edge picks a smart response and sends the contents of the web page that corresponds to that smart response to the kid-client. If the URL is not in either list, the edge connects to the server which hosts the web pages corresponding to the URL, gets the contents of the web pages, and analyses them with the help of the AI classifiers. If the content is suitable for that child, the edge forwards it to the kid-client. Otherwise, a smart response is forwarded to the kid-client. Then, the kid-client redirects the contents to the web browser. Finally, the edge server sends the results of the AI check to the operator, and the operator sends possible updates to the allow/deny list to all edge servers.

Please note that by utilizing a general check, the web pages that a certain child is interested in are kept private from the service providers, because the contents of the web pages are always sent to the edge servers (not directly to the child's device).

In the off-line phase of the personalized check about accessing a web page, the parent writes the attributes that they do not want their child to be exposed to in the parent-client application. The parent-client inserts the parent's attributes in a set W_p.

The website that the child of this parent is interested in has a set of attributes that is denoted by W_c. The set W_c is obtained by the edge server that is handling the child's traffic (say, Edge server 1). Then, the child should be able to access this website only if the intersection of sets W_p and W_c is an empty set.

As we explained before, we want to preserve the privacy of the people that are involved in our protocols. In the parental control protocols, we want to preserve the privacy of parents and children towards each other and also towards the service providers and the network. Therefore, the set $W_p \cap W_c$ should be computed in a private manner. Moreover, it should not be the parent-client nor edge server who learns the output set $W_p \cap W_c$, but instead an external decider should learn this output. We choose the kid-client as the external decider[5], and use the PSO-Lim protocol of Subsection 5.1.1 to compute the outcome set $W_p \cap W_c$. Please note that the set of all possible attributes for a web page is limited, and therefore the operator can generate an ordered set U to represent all of these attributes. The operator sends the set U to all the edge servers and parent-clients in the off-line phase of the protocol. Still in the off-line phase, the operator sends several addresses of other edge servers than Edge 1 to the kid-client. The parent-client performs its part of computations for the PSO-Lim protocol on the set W_p, and sends the results to Edge 1.

In the on-line phase, similarly to the general check, the kid wants to access a website. The web browser sends the URL corresponding to the child's desired website to the kid-client. Then, the kid-client sends the URL to Edge server 1, and the edge server looks into the deny list for this URL[6]. If the URL is in the deny list, the edge picks a smart response and sends the corresponding web page to the kid-client, and consequently to the web browser (via the kid-client) and the protocol stops. Otherwise, Edge 1 retrieves the contents on web pages corresponding to this URL, analyses the contents and creates the set W_c. Edge server 1 performs the computations of the PSO-Lim protocol, with the results that the parent-client sent in the off-line phase and the set W_c, and sends the results to the kid-client. The edge also picks a URL from the smart responses and sends it to the kid-client.

[5] Although, kid-client is a party in our parental control protocol, it is not a party with input set in the PSI protocol that is required to obtain $W_p \cap W_c$. Therefore, the kid-client (not the child) is an external decider in this PSI protocol.

[6] Please note that in the personalized check, the edge server does not check the URL against the allow list. This is because, even if a URL is in the allow list, it might contain features that the parent wishes to block.

The kid-client receives this result from Edge 1, acts as the external decider, decrypts the result, and learns whether or not $W_p \cap W_c$ is empty. If the result is an empty set, kid-client picks an address[7] of an edge server (say, Edge 2), and sends the URL to this edge server. Otherwise, kid-client sends the smart response URL that it got earlier from Edge 1, to Edge 2. Next, Edge 2 sends the contents of the requested web pages to the kid-client, and the kid-client redirects the contents to the web browser. Figure 6.3 shows a summary of the personalized check protocol for accessing a web page.

Please note that the original URL or the smart response is sent to another edge server than Edge 1 to protect the privacy of the parent. Also, note that the kid who receives a smart response does not learn whether it was the operator or the parent that blocked their access to the original URL. Lastly, the parent does not learn which web pages their kid is interested in and therefore the privacy of the child towards their parent is preserved.

Parental Control via Protected Text Messaging

In this section, we present two parental control protocols for protected text messaging; the general check protocol and the personalized check protocol. These protocols are presented in Publication VII.

In the set-up phase of our general check protocol to detect harmful content in a text message, the operator collects a list of harmful keywords and stores them in a set B. The operator sends the set B to all edge servers. In the on-line phase, the kid writes a message that consists of one or several words. The kid-client stores the words in a set M. The edge server and the kid-client enter their respective sets B and M into a PSI protocol. If the result of the PSI is an empty set, the message is forwarded to its receiver. Otherwise, the kid-client sends the message to the edge server, where the message is checked for traces of harmful content with the help of AI. If the check shows that the message is benign, it is forwarded. Otherwise, a smart response is sent to the child that created the original message. Here, a smart response could be, e.g., a message that motivates and encourages the child to use kid-friendly words. For more details about smart responses please see Publication VII.

We explained earlier that the current parental control applications are designed in such a way that the parents are getting copies of the text messages that their children sent or received. In this subsection, we present a privacy-preserving protocol that enables parents to protect their children

[7] As mentioned before, in the off-line phase the operator sent several addresses of other edge servers than Edge 1 to the kid-client.

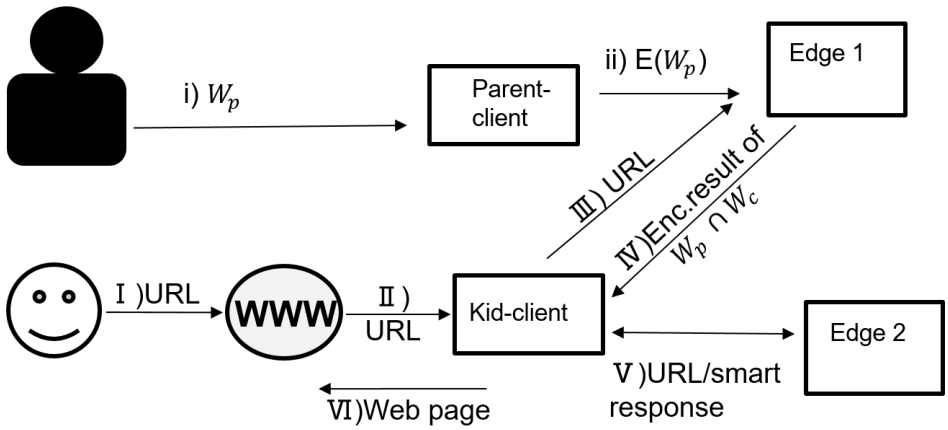


Figure 6.3: An overview of the off-line and on-line phases of the personalized check in our parental control protocol. The steps of off-line (on-line, respectively) phase are marked with lower-case (upper-case, respectively) roman numbers.

against harmful text messages, without learning the child's message. Similarly to the protected web-surfing, we utilize two edge servers (Edge 1 and Edge 2) in the protected text messaging protocol to handle the child's traffic so that neither Edge server 1 nor Edge server 2 learn the parent's private inputs, nor the outcome of the protocol. Our personalized check protocol is as follows.

In the set-up phase, the operator creates the set B and sends it to all the edge servers, as explained in the general check protocol. In the off-line phase, the parent creates a set of forbidden words L, and two sets of allow and deny contact numbers, and inserts these three sets into the parent-client.

In the on-line phase, when the child enters the name of the recipient of the message in the messaging app, the application sends the receiver's contact number to the edge. Then the kid-client (the external decider), the edge and the parent-client together execute two PMT protocols where the kid-client will learn whether the edge's item (the contact number) is in any of the allow and deny sets of the parent-client[8]. If the receiver is in the deny (allow, respectively) list, the kid-client blocks the message (redirects it to Edge server 2, respectively). Otherwise, the kid-client inserts the words in the message into a set M. The kid-client, parent-client, Edge 1,

[8]One example of such a PMT protocol is the PSO-UnLim protocol of Subsection 5.1.1 where one of the sets is a singleton.

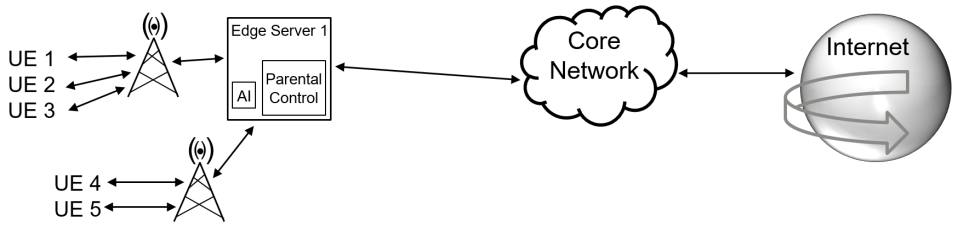


Figure 6.4: The network architecture with an edge server that contains AI-assisted parental control functionalities.

and Edge 2 execute a PSO protocol to determine the cardinality of the set $M \cap (B \cup L)$. This cardinality is only learnt by Edge server 2. Please note that in this PSO protocol, the kid-client has an input set and therefore is not an external decider, but instead Edge server 2 fits our definition for an external decider. We can use the PSO-UnLim protocol of Subsection 5.1.1 to retrieve the cardinality of the output set.

If $M \cap (B \cup L)$ is an empty set, the message is clean and can be directed to its recipient. Otherwise, the message is blocked and Edge server 2 sends a smart response text message to the sender.

Figure 6.4 envisions the network architecture in the context of parental control. Please note that our parental control protocols can be altered such that they can be used for some other application areas, where a centralized privacy preserving protocol is needed. For example, protecting the elderly from email phishing in a privacy preserving way.

Chapter 7

Discussions and Conclusion

With ever-progressing development of digital technologies such as mobile devices and on-line games, concerns about how to protect these technologies are increasing as well. Moreover, especially after the COVID-19 world pandemic [159], the usage of internet and internet dependency in everyday life are increasing [160]. This means the importance of protecting the privacy and security of internet users and also the networking environments is realized now more than ever [161, 162].

In this dissertation, we focused on developing privacy-preserving protocols for protected networking. We aimed to develop our protocols in such a way that the users can get the benefits of digital worlds in a private and secure manner, without sacrificing communication time and bandwidth usage.

The dissertation is based on the eight publications, Publications I - VIII, that are listed on pages ix and x. Next, we give a summary of the findings and protocols of this dissertation:

- Design of a general private set operation protocol with an external decider that can be used when the universe is limited.
- Design of a general private set operation protocol with an external decider to retrieve the cardinality of the outcome set. This protocol can also be used when the universe is not limited.
- Design of four private membership test protocols.
- Design of several privacy-preserving protocols for graph queries.

Designing privacy-preserving methods can be applied to a variety of application areas. We delved deeper into the following three areas: i) malware protection, ii) protection of remote access, and iii) protecting minors.

The main findings of this research regarding these application areas are as follows:

- We designed privacy-preserving malware check protocols in a cloud environment.
- We designed privacy-preserving protocols to check remote access chains.
- We designed parental control protocols and cyberbullying prevention protocols in 5G networks.

Publication I and Publication II provide PMT protocols that can be used in malware protection applications, among other things. The protocols of Publication I are suitable for the case scenarios where the time complexity should be minimal, while at the same time keeping the communication costs on a feasible level. The protocol of Publication II keeps the communication complexity low, without losing the practicality of the protocol regarding time consumption. In other words, both publications propose feasible PMT protocols. The protocols of Publication I have lower time complexity than the protocol of Publication II. The protocol of Publication II has lower communication complexity than the protocols of Publication I. Future work can focus on designing a PMT protocol that lowers both time and communication costs.

The PMT protocol is a special case of a PSO protocol. Publication VIII provides a general PSO protocol when the members of the private sets are all in a limited universe, and for the case where an external decider is part of the protocol. We also present a general PSO protocol to discover the cardinality of the output set, with the presence of an external decider. One direction for future work can be to design a general PSO protocol with an external decider.

In Publications III - V, we present privacy-preserving query protocols on graphs. One use-case of these protocols is to utilize them to query the database of remote access privileges. Our work can be extended by designing privacy-preserving protocols that can perform queries on a graph that is owned by more than two owners.

Publication VI presents privacy-preserving cyberbullying prevention protocols, and Publication VII presents privacy-preserving parental control protocols. The importance of the parental control and cyberbullying prevention techniques comes from the fact that children are the most vulnerable users in the internet. The protocols of Publications VI and VII describe how to provide privacy and security for all children, regardless of whether their parents are involved with their child's digital life or not. Therefore,

provide an equal opportunity for all children to experience their digital life. As a future work, we suggest to further improve our minor protection protocols and develop them to standard modules in the future generations of mobile networks.

We mentioned earlier that the COVID-19 pandemic caused an increase in our everyday life dependency on the internet. Some studies suggest that this increase in utilizing the internet has several negative impacts on the users, specially on children [163, 164]. Also the COVID-19 pandemic emphasized the importance of other problems that we studied. There has been a significant increase in the number of digital remote accesses [165], as well as attackers abusing the pandemic to encourage users to click on links that look benign but contain malware [166]. Therefore, the importance of Publications I - VII has been increased after the COVID-19 pandemic.

References

[1] Sara Ramezanian, Tommi Meskanen, Masoud Naderpour, and Valtteri Niemi. Private membership test protocol with low communication complexity. In *Proceedings of the International Conference on Network and System Security*, pages 31–45. Springer, 2017.

[2] Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. Privacy preserving shortest path queries on directed graph. In *Proceedings of the 22nd Conference of Open Innovations Association (FRUCT)*, pages 217–223. IEEE, 2018.

[3] Sara Ramezanian and Valtteri Niemi. Privacy preserving cyberbullying prevention with AI methods in 5G networks. In *Proceedings of the 25th Conference of Open Innovations Association (FRUCT)*, pages 265–271. IEEE, 2019.

[4] Sonia Livingstone, Leslie Haddon, Anke Görzig, and Kjartan Ólafsson. Risks and safety on the internet. *The Perspective of European Children. Full Findings and Policy Implications from the EU Kids Online Survey of 9-16 Year Olds and Their Parents in 25 Countries*, pages 9–16, 2011.

[5] Yubo Hou, Dan Xiong, Tonglin Jiang, Lily Song, and Qi Wang. Social media addiction: Its impact, mediation, and intervention. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 13(1), 2019.

[6] Erin A Vogel, Jason P Rose, Lindsay R Roberts, and Katheryn Eckles. Social comparison, social media, and self-esteem. *Psychology of Popular Media Culture*, 3(4):206, 2014.

[7] Waseem Akram and Rekesh Kumar. A study on positive and negative effects of social media on society. *International Journal of Computer Sciences and Engineering*, 5(10):351–354, 2017.

[8] Winnie Chung and John Paynter. Privacy issues on the internet. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 9–pp. IEEE, 2002.

[9] Karen D Loch, Houston H Carr, and Merrill E Warkentin. Threats to information systems: Today's reality, yesterday's understanding. *Mis Quarterly*, pages 173–186, 1992.

[10] Lee Rainie and Maeve Duggan. *Privacy and Information Sharing*. Pew Research Center, 2016.

[11] Cristian Bravo-Lillo, Lorrie Faith Cranor, Julie Downs, and Saranga Komanduri. Bridging the gap in computer security warnings: A mental model approach. *IEEE Security & Privacy*, 9(2):18–26, 2010.

[12] Suranga Seneviratne, Aruna Seneviratne, Prasant Mohapatra, and Anirban Mahanti. Predicting user traits from a snapshot of apps installed on a smartphone. *ACM SIGMOBILE Mobile Computing and Communications Review*, 18(2):1–8, 2014.

[13] Lawrence G Roberts and Barry D Wessler. Computer network development to achieve resource sharing. In *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference*, pages 543–549, 1970.

[14] Gerald A Marin. Network security basics. *IEEE Security & Privacy*, 3(6):68–72, 2005.

[15] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2018.

[16] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*, volume 19. John Wiley & Sons, 2004.

[17] Linda Rosencrance. Remote access. `https://searchsecurity.techtarget.com/definition/remote-access`, 2021. Accessed: 2021-10-23.

[18] Oxford Learners' Dictionaries. Security. `https://www.oxfordlearnersdictionaries.com/definition/english/security?q=security`, 2021. Accessed: 2021-10-05.

[19] Gabriele Piccoli and Federico Pigni. *Information Systems for Managers: with Cases*. Prospect Press, 2019.

[20] Charles Parker and Thomas L Case. *Management Information Systems: Strategy and Action*. McGraw-Hill Education, 1993.

[21] Mikko T Siponen. An analysis of the traditional is security approaches: Implications for research and practice. *European Journal of Information Systems*, 14(3):303–315, 2005.

[22] Ed Crowley. Information system security curricula development. In *Proceedings of the 4th Conference on Information Technology Curriculum*, pages 249–255, 2003.

[23] Daniel J Solove. Conceptualizing privacy. *California Law Review*, 90:1087, 2002.

[24] Serge Gutwirth. *Privacy and the Information Age*. Rowman & Littlefield, 2002.

[25] Paul Sieghart. *Privacy and Computers*. Latimer New Dimensions, 1976.

[26] Colin J Bennett. *Regulating Privacy: Data Protection and Public Policy in Europe and the United States*. Cornell University Press, 1992.

[27] Rachel L Finn, David Wright, and Michael Friedewald. Seven types of privacy. In *European Data Protection: Coming of Age*, pages 3–32. Springer, 2013.

[28] Richard A Posner. The right of privacy. *Georgia Law Review*, 12:393, 1977.

[29] Tony J Perri. Who wants privacy protection, and what do they want? *Journal of Consumer Behaviour: An International Research Review*, 2(1):80–100, 2002.

[30] Paul E Black. Data structure. `https://www.nist.gov/dads/HTML/dataStructure.html`, 2004. Accessed: 2021-08-14.

[31] Peter Wegner and Edwin D Reilly. *Data Structures*. John Wiley and Sons Ltd., GBR, 2003.

[32] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[33] Lailong Luo, Deke Guo, Richard TB Ma, Ori Rottenstreich, and Xueshan Luo. Optimizing bloom filter: Challenges, solutions, and comparisons. *IEEE Communications Surveys & Tutorials*, 21(2):1912–1949, 2018.

[34] Bin Fan, Dave G Andersen, Michael Kaminsky, and Michael D Mitzenmacher. Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 75–88, 2014.

[35] Md Saidur Rahman. *Basic Graph Theory*. Springer, 2017.

[36] Roger C Schank. What is ai, anyway? *AI Magazine*, 8(4):59–59, 1987.

[37] Linda S Gottfredson. Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography, 1997.

[38] John McCarthy. What is artificial intelligence? *Computer Science Department, Stanford University*, 2007.

[39] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

[40] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[41] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[42] Nils J Nilsson. *Learning Machines*. McGrawHill: New York, 1965.

[43] Roshan Kumari and Saurabh Kr Srivastava. Machine learning: A review on binary classification. *International Journal of Computer Applications*, 160(7), 2017.

[44] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.

[45] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2020.

[46] Pankaj Deep Kaur and Inderveer Chana. Unfolding the distributed computing paradigms. In *Proceedings of the International Conference on Advances in Computer Engineering*, pages 339–342. IEEE, 2010.

[47] Ramnath Chellappa. Intermediaries in cloud-computing: A new computing paradigm. In *INFORMS Annual Meeting, Dallas*, pages 26–29, 1997.

[48] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[49] Theodore S Rappaport. *Wireless Communications: Principles and Practice*, volume 2. Prentice Hall PTR New Jersey, 1996.

[50] Akhil Gupta and Rakesh K Jha. A survey of 5G network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.

[51] 3GPP. 3GPP TS 23.501 - System Architecture for the 5G System. `https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/`, 2020. Accessed: 2020-08-17.

[52] Andrew C Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 160–164. IEEE, 1982.

[53] Wenliang Du and Mikhail J Atallah. Secure multi-party computation problems and their applications: A review and open problems. In *Proceedings of the 2001 Workshop on New Security Paradigms*, pages 13–22, 2001.

[54] Ronald Cramer and Ivan Damgård. Multiparty computation, an introduction. In *Contemporary Cryptology*, pages 41–87. Springer, 2005.

[55] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: Theory, practice and applications. *Information Sciences*, 476:357–372, 2019.

[56] Benny Pinkas, Thomas Schneider, and Michael Zohner. Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS)*, 21(2):1–35, 2018.

[57] Sandeep Tamrakar, Jian Liu, Andrew Paverd, Jan-Erik Ekberg, Benny Pinkas, and N Asokan. The circle game: Scalable private membership test using trusted hardware. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 31–44, 2017.

[58] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 4(11):169–180, 1978.

[59] Monique Ogburn, Claude Turner, and Pushkar Dahal. Homomorphic encryption. *Procedia Computer Science*, 20:502–509, 2013.

[60] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 365–77, 1982.

[61] Pascal Paillier and David Pointcheval. Efficient public-key cryptosystems provably secure against active adversaries. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pages 165–179. Springer, 1999.

[62] Benny Chor, Niv Gilboa, and Moni Naor. *Private Information Retrieval by Keywords*. Citeseer, 1997.

[63] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 364–373. IEEE, 1997.

[64] William Gasarch. A survey on private information retrieval. *Bulletin of the EATCS*, 82:72–107, 2004.

[65] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pages 803–815. Springer, 2005.

[66] Yan-Cheng Chang. Single database private information retrieval with logarithmic communication. In *Proceedings of the Australasian Conference on Information Security and Privacy*, pages 50–61. Springer, 2004.

[67] André Nies. Differentiability of polynomial time computable functions. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[68] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[69] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[70] Michael J Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Proceedings of the Theory of Cryptography Conference*, pages 303–324. Springer, 2005.

[71] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Proceedings of the International Workshop on Fast Software Encryption*, pages 371–388. Springer, 2004.

[72] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology*, pages 199–203. Springer, 1983.

[73] Carmit Hazay and Yehuda Lindell. A note on the relation between the definitions of security for semi-honest and malicious adversaries. *The International Association for Cryptologic Research (IACR) ePrint Archive*, 2010:551, 2010.

[74] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[75] Iliano Cervesato. The dolev-yao intruder is the most powerful attacker. In *16th Annual Symposium on Logic in Computer Science (LICS)*, volume 1. Citeseer, 2001.

[76] Joseph Y Halpern and Riccardo Pucella. Modeling adversaries in a logic for security protocol analysis. In *Formal Aspects of Security*, pages 115–132. Springer, 2002.

[77] Mohamed Abomhara and Geir M Køien. Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, pages 65–88, 2015.

[78] Carmit Hazay and Yehuda Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer Science & Business Media, 2010.

[79] Nwokedi Idika and Aditya P Mathur. A survey of malware detection techniques. *Purdue University*, 48:2007–2, 2007.

[80] Loick Menvielle, Anne-Françoise Audrain-Pontevia, and William Menvielle. *The Digitization of Healthcare: New Challenges and Opportunities*. Springer, 2017.

[81] Oliver Ahel and Katharina Lingenau. Opportunities and challenges of digitalization to improve access to education for sustainable development in higher education. In *Universities as Living Labs for Sustainable Development*, pages 341–356. Springer, 2020.

[82] Paul Davidsson, Banafsheh Hajinasab, Johan Holmgren, Åse Jevinger, and Jan A Persson. The fourth wave of digitalization and public transport: Opportunities and challenges. *Sustainability*, 8(12):1248, 2016.

[83] Chakravanti Rajagopalachari Kothari. *Research Methodology: Methods and Techniques*. New Age International, 2004.

[84] Bill Bailey. Case studies: A security science research methodology. 2011.

[85] Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339, 2019.

[86] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2008.

[87] Gareth Morgan and Linda Smircich. The case for qualitative research. *Academy of Management Review*, 5(4):491–500, 1980.

[88] Yaniv Erlich and Arvind Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409–421, 2014.

[89] Mona FM Mursi, Ghazy MR Assassa, Ahmed Abdelhafez, and Kareem M Abo Samra. On the development of electronic voting: A survey. *International Journal of Computer Applications*, 61(16), 2013.

[90] Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. Pir-psi: Scaling private contact discovery. *Proceedings of Privacy Enhancing Technologies*, 2018(4):159–178, 2018.

[91] Shishir Nagaraja, Prateek Mittal, Chi-Yao Hong, Matthew Caesar, and Nikita Borisov. Botgrep: Finding p2p bots with structured graph analysis. In *Proceedings of the USENIX Security Symposium*, volume 10, pages 95–110, 2010.

[92] Manas A Pathak, Bhiksha Raj, Shantanu D Rane, and Paris Smaragdis. Privacy-preserving speech processing: Cryptographic and string-matching frameworks show promise. *IEEE Signal Processing Magazine*, 30(2):62–74, 2013.

[93] Emiliano De Cristofaro and Gene Tsudik. Experimenting with fast private set intersection. In *Proceedings of the International Conference on Trust and Trustworthy Computing*, pages 55–73. Springer, 2012.

[94] Aydin Abadi, Sotirios Terzis, and Changyu Dong. O-psi: Delegated private set intersection on outsourced datasets. In *Proceedings of the IFIP International Information Security and Privacy Conference*, pages 3–17. Springer, 2015.

[95] Giuseppe Ateniese, Emiliano De Cristofaro, and Gene Tsudik. (if) size matters: Size-hiding private set intersection. In *Proceedings of the International Workshop on Public Key Cryptography*, pages 156–173. Springer, 2011.

[96] Changyu Dong, Liqun Chen, and Zikai Wen. When private set intersection meets big data: An efficient and scalable protocol. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pages 789–800, 2013.

[97] Lea Kissner and Dawn Song. Privacy-preserving set operations. In *Proceedings of the Annual International Cryptology Conference*, pages 241–257. Springer, 2005.

[98] Keith Frikken. Privacy-preserving set union. In *Proceedings of the International Conference on Applied Cryptography and Network Security*, pages 237–252. Springer, 2007.

[99] Ji Young Chun, Dowon Hong, Ik Rae Jeong, and Dong Hoon Lee. Privacy-preserving disjunctive normal form operations on distributed sets. *Information Sciences*, 231:113–122, 2013.

[100] Wenli Wang, Shundong Li, Jiawei Dou, and Runmeng Du. Privacy-preserving mixed set operations. *Information Sciences*, 525:67–81, 2020.

[101] Uri Feige, Joe Killian, and Moni Naor. A minimal model for secure computation. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 554–563, 1994.

[102] Benny Pinkas, Thomas Schneider, and Michael Zohner. Faster private set intersection based on OT extension. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)*, pages 797–812, 2014.

[103] Anunay Kulshrestha and Jonathan Mayer. Identifying harmful media in end-to-end encrypted communication: Efficient private membership computation. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[104] Eduardo Chielle, Homer Gamil, and Michail Maniatakos. Real-time private membership test using homomorphic encryption. In *Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1282–1287. IEEE, 2021.

[105] Fucai Zhou, Zifeng Xu, Yuxi Li, Jian Xu, and Su Peng. Private graph intersection protocol. In *Proceedings of the Australasian Conference on Information Security and Privacy*, pages 235–248. Springer, 2017.

[106] Xiangjian Zuo, Lixiang Li, Shoushan Luo, Haipeng Peng, Yixian Yang, and Linming Gong. Privacy-preserving verifiable graph intersection scheme with cryptographic accumulators in social networks. *IEEE Internet of Things Journal*, 8(6):4590–4603, 2020.

[107] Justin Brickell and Vitaly Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pages 236–252. Springer, 2005.

[108] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 329–340. Springer, 2013.

[109] Ghita Mezzour, Adrian Perrig, Virgil Gligor, and Panos Papadimitratos. Privacy-preserving relationship path discovery in social networks. In *Proceedings of the International Conference on Cryptology and Network Security*, pages 189–208. Springer, 2009.

[110] Kun Liu, Kamalika Das, Tyrone Grandison, and Hillol Kargupta. Privacy-preserving data analysis on graphs and social networks. In *Next generation of data mining*, pages 443–462. Chapman and Hall/CRC, 2008.

[111] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *The Structure and Dynamics of Networks*, pages 183–194. Princeton University Press, 2011.

[112] Mingwu Zhang, Yu Chen, and Willy Susilo. Ppo-cpq: A privacy-preserving optimization of clinical pathway query for e-healthcare systems. *IEEE Internet of Things Journal*, 7(10):10660–10672, 2020.

[113] Yong Xi, Loren Schwiebert, and Weisong Shi. Privacy preserving shortest path routing with an application to navigation. *Pervasive and Mobile Computing*, 13:142–149, 2014.

[114] David J Wu, Joe Zimmerman, Jérémy Planul, and John C Mitchell. Privacy-preserving shortest path computation. *arXiv preprint arXiv:1601.02281*, 2016.

[115] Chang Liu, Liehuang Zhu, Xiangjian He, and Jinjun Chen. Enabling privacy-preserving shortest distance queries on encrypted graph data. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[116] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

[117] Wei-Ta Chu and Feng-Chi Chang. A privacy-preserving bipartite graph matching framework for multimedia analysis and retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 243–250, 2015.

[118] Jian Zhou, Jiwu Jing, Ji Xiang, and Lei Wang. Privacy preserving social network publication on bipartite graphs. In *Proceedings of the IFIP International Workshop on Information Security Theory and Practice*, pages 58–70. Springer, 2012.

[119] Tai-Hoon Kim, Gulshan Kumar, Rahul Saha, Mritunjay Kumar Rai, William J Buchanan, Reji Thomas, and Mamoun Alazab. A privacy preserving distributed ledger framework for global human resource record management: The blockchain aspect. *IEEE Access*, 8:96455–96467, 2020.

[120] Salasiah Abdullah and Khairul Azmi Abu Bakar. Security and privacy challenges in cloud computing. In *Proceedings of the 2018 Cyber Resilience Conference (CRC)*, pages 1–3. IEEE, 2018.

[121] Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu. Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks*, 10(7):190903, 2014.

[122] Jiale Zhang, Bing Chen, Yanchao Zhao, Xiang Cheng, and Feng Hu. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. *IEEE Access*, 6:18209–18237, 2018.

[123] Gary McGraw and Greg Morrisett. Attacking malicious code: A report to the infosec research council. *IEEE Software*, 17(5):33–41, 2000.

[124] Amit Vasudevan and Ramesh Yerraballi. Spike: Engineering malware analysis tools using unobtrusive binary-instrumentation. In *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, pages 311–320. Citeseer, 2006.

[125] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy*, 8(6):24–31, 2010.

[126] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011.

[127] Zhifeng Xiao and Yang Xiao. Security and privacy in cloud computing. *IEEE Communications Surveys & Tutorials*, 15(2):843–859, 2012.

[128] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou. Security and privacy in cloud computing: A survey. In *Proceedings of the Sixth International Conference on Semantics, Knowledge and Grids*, pages 105–112. IEEE, 2010.

[129] Paolo Palumbo, Alexey Kirichenko, Valtteri Niemi, Sara Ramezanian, and Tommi Meskanen. Method for integrity protection in a computer network, November 26 2020. US Patent App. 16/878,312.

[130] George E Dahl, Jack W Stokes, Li Deng, and Dong Yu. Large-scale malware classification using random projections and neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3422–3426. IEEE, 2013.

[131] Yanfang Ye, Tao Li, Donald Adjeroh, and S Sitharama Iyengar. A survey on malware detection using data mining techniques. *ACM Computing Surveys (CSUR)*, 50(3):1–40, 2017.

[132] Matthias Jarke and Jurgen Koch. Query optimization in database systems. *ACM Computing Surveys (CsUR)*, 16(2):111–152, 1984.

[133] Raghu Ramakrishnan, Johannes Gehrke, and Johannes Gehrke. *Database Management Systems*, volume 3. McGraw-Hill New York, 2003.

[134] Edgar F Codd. Data models in database management. In *Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modeling*, pages 112–114, 1980.

[135] Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1):1–39, 2008.

[136] Andrew V Goldberg and Chris Harrelson. Computing the shortest path: A search meets graph theory. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 156–165, 2005.

[137] F Benjamin Zhan. Three fastest shortest path algorithms on real road networks: Data structures and procedures. *Journal of Geographic Information and Decision Analysis*, 1(1):69–82, 1997.

[138] Kai Wang, Wenjie Zhang, Xuemin Lin, Ying Zhang, Lu Qin, and Yuting Zhang. Efficient and effective community search on large-scale bipartite graphs. In *Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE)*, pages 85–96. IEEE, 2021.

[139] Vicky Rideout. *The Common Sense Census: Media Use by Tweens and Teens*. San Francisco, CA: Common Sense Media, 2015.

[140] Jonathan Y Bernard, Natarajan Padmapriya, Bozhi Chen, Shirong Cai, Kok Hian Tan, Fabian Yap, Lynette Shek, Yap-Seng Chong, Peter D Gluckman, Keith M Godfrey, Michael S Kramer, Seang M Saw, and Falk Muller-Riemenschneider. Predictors of screen viewing time in young Singaporean children: the GUSTO cohort. *International Journal of Behavioral Nutrition and Physical Activity*, 14(1):1–10, 2017.

[141] Sonia Livingstone and Ellen J Helsper. Parental mediation of children's internet use. *Journal of Broadcasting & Electronic Media*, 52(4):581–599, 2008.

[142] Chang-Hoan Cho and Hongsik John Cheon. Children's exposure to negative internet content: Effects of family context. *Journal of Broadcasting & Electronic Media*, 49(4):488–509, 2005.

[143] Patti M Valkenburg and Karen E Soeters. Children's positive and negative experiences with the internet: an exploratory survey. *Communication research*, 28(5):652–675, 2001.

[144] Abdul Razaque Chhachhar, Barkatullah Qureshi, Zulfiqar Ahmed Maher, and Shakil Ahmed. Influence of internet websites on children study. *Journal of American Science*, 10(5):40–45, 2014.

[145] Walter Fuertes, Karina Quimbiulco, Fernando Galárraga, and José Luis García-Dorado. On the development of advanced parental control tools. In *Proceedings of the 1st International Conference on Software Security and Assurance (ICSSA)*, pages 1–6. IEEE, 2015.

[146] Sharifa Alghowinem. A safer youtube kids: An extra layer of content filtering using automated multimodal analysis. In *Proceedings of SAI Intelligent Systems Conference*, pages 294–308. Springer, 2018.

[147] Otávio de P Albuquerque, Marcelo Fantinato, Marcelo M Eler, Sarajane M Peres, and Patrick CK Hung. A study of parental control requirements for smart toys. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2215–2220. IEEE, 2020.

[148] Skyler T Hawk, William W Hale III, Quinten AW Raaijmakers, and Wim Meeus. Adolescents' perceptions of privacy invasion in reaction to parental solicitation and control. *The Journal of Early Adolescence*, 28(4):583–608, 2008.

[149] Arup K Ghosh, Karla Badillo-Urquiola, Shion Guha, Joseph J LaViola Jr, and Pamela J Wisniewski. Safety vs. surveillance: What children have to say about mobile apps for parental control. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[150] Emmanouil Magkos, Eleni Kleisiari, Panagiotis Chanias, and Viktor Giannakouris-Salalidis. Parental control and children's internet safety: the good, the bad and the ugly. In *Proceedings of the International Conference on Industrial Logistics (ICIL)*, pages 829–848, 2014.

[151] Suzan Ali, Mounir Elgharabawy, Quentin Duchaussoy, Mohammad Mannan, and Amr Youssef. Betrayed by the guardian: Security and privacy risks of parental control solutions. In *Proceedings of the Annual Computer Security Applications Conference*, pages 69–83, 2020.

[152] Suzan Ali, Mounir Elgharabawy, Quentin Duchaussoy, Mohammad Mannan, and Amr Youssef. Parental controls: Safer internet solutions or new pitfalls? *IEEE Security & Privacy*, pages 36–46, 2021.

[153] Kirk R Williams and Nancy G Guerra. Prevalence and predictors of internet bullying. *Journal of Adolescent Health*, 41(6):S14–S21, 2007.

[154] Nancy E Willard. *Cyberbullying and Cyberthreats: Responding to the Challenge of Online Social Aggression, Threats, and Distress*. Research Press, 2007.

[155] Dieter Wolke, William E Copeland, Adrian Angold, and E Jane Costello. Impact of bullying in childhood on adult health, wealth, crime, and social outcomes. *Psychological Science*, 24(10):1958–1970, 2013.

[156] Rachel C Vreeman and Aaron E Carroll. A systematic review of school-based interventions to prevent bullying. *Archives of Pediatrics & Adolescent Medicine*, 161(1):78–88, 2007.

[157] 3GPP. Service requirements for the evolved packet system (eps). `https://www.3gpp.org/ftp/Specs/archive/22_series/22.278/`, 2019. Accessed: 2020-08-17.

[158] Mahdi Hashemi. Web page classification: A survey of perspectives, gaps, and future directions. *Multimedia Tools and Applications*, pages 1–25, 2020.

[159] Marco Ciotti, Massimo Ciccozzi, Alessandro Terrinoni, Wen-Can Jiang, Cheng-Bin Wang, and Sergio Bernardini. The Covid-19 pandemic. *Critical Reviews in Clinical Laboratory Sciences*, 57(6):365–388, 2020.

[160] Rahul De, Neena Pandey, and Abhipsa Pal. Impact of digital surge during covid-19 pandemic: A viewpoint on research and practice. *International Journal of Information Management*, 55:102171, 2020.

[161] Aaron R Brough and Kelly D Martin. Consumer privacy during (and after) the Covid-19 pandemic. *Journal of Public Policy & Marketing*, 40(1):108–110, 2021.

[162] Navid A Khan, Sarfraz N Brohi, and Noor Zaman. Ten deadly cyber security threats amid Covid-19 pandemic. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.12278792.v1, 2020.

[163] Suely F Deslandes and Tiago Coutinho. The intensive use of the internet by children and adolescents in the context of Covid-19 and the risks for self-inflicted violence. *Ciencia & Saude Coletiva*, 25:2479–2486, 2020.

[164] Huixi Dong, Fangru Yang, Xiaozi Lu, and Wei Hao. Internet addiction and related psychological factors among children and adolescents in China during the coronavirus disease 2019 (Covid-19) epidemic. *Frontiers in Psychiatry*, 11:751, 2020.

[165] Thomas Favale, Francesca Soro, Martino Trevisan, Idilio Drago, and Marco Mellia. Campus traffic and e-learning during Covid-19 pandemic. *Computer Networks*, 176:107290, 2020.

[166] Bernardi Pranggono and Abdullahi Arabo. Covid-19 pandemic cybersecurity issues. *Internet Technology Letters*, 4(2):e247, 2021.