DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS
77

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS
77

BINGSHENG ZHANG

Efficient cryptographic protocols for secure and private remote databases

TARTU UNIVERSITY PRESS

Institute of Computer Science, Faculty of Mathematics and Computer Science, University of Tartu, Estonia

Dissertation accepted for public defense of the degree of Doctor of Philosophy (PhD) on 30 August, 2011 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors:

Dr.	Helger Lipmaa University of Tartu Tartu, Estonia
Prof.	Peeter Laud University of Tartu Tartu, Estonia

Opponents:

Dr.	Jens Groth University College London London, UK
Prof.	Jesper Buus Nielsen University of Aarhus Aarhus, Denmark

The public defense will take place on October 10, 2011 at 16:15 in Liivi 2-405.

The publication of this dissertation was financed by Institute of Computer Science, University of Tartu.

ISSN 1024-4212
ISBN 978-9949-19-845-0 (trükis)
ISBN 978-9949-19-846-7 (PDF)

Tartu Ülikooli Kirjastus
http://www.tyk.ee
Tellimus nr. 572

Contents

LIST OF ORIGINAL PUBLICATIONS

1. Lipmaa, H., Zhang, B.: Efficient Generalized Selective Private Function Evaluation with Applications in Biometric Authentication. In: Proceedings of the 5th Information Security and Cryptology, Inscrypt '09. LNCS, vol. 6151, pp. 154–163. Springer (2009).

2. Lipmaa, H., Zhang, B.: Two New Efficient PIR-Writing Protocols. In: Proceedings of the 8th International Conference on Applied cryptography and network Security, ACNS '10. LNCS, vol. 6123, pp. 438–455. Springer (2010).

3. Zhang, B.: Generic Constant-Round Oblivious Sorting Algorithm for MPC. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 240–256. Springer (2011).

4. Laur, S., Willemson, J., Zhang, B.: Round-efficient Oblivious Database Manipulation. In: Proceedings of the 14th Information Security Conference, ISC '11. LNCS, Springer (2011).

5. Zhang, B.: Simulatable Adaptive Oblivious Transfer With Statistical Receiver's Privacy. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 52–67. Springer (2011).

UNPUBLISHED WORK INCLUDED IN THE THESIS

6. Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Cryptology ePrint Archive, Report 2011/394 (2011), `http://eprint.iacr.org/2011/394`.

PUBLICATIONS NOT INCLUDED IN THE THESIS

7. Nakahara, J., Sepehrdad, P., Zhang, B., Wang, M.: Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In: Proceedings of the 8th International Conference on Cryptology and Network Security, CANS '09. LNCS, vol. 5888, pp. 58–75. Springer-Verlag, Berlin, Heidelberg (2009).

8. Bard, G.V., Courtois, N., Nakahara, J., Sepehrdad, P., Zhang, B.: Algebraic, AIDA/Cube and Side Channel Analysis of KATAN Family of Block Ciphers. In: Proceedings of the 11th International Conference on Cryptology in India, INDOCRYPT '10. LNCS, vol. 6498, pp. 176–196. Springer (2010).

ABSTRACT

Nowadays, "Cloud computing" is being heavily promoted by market leaders such as Apple, Google and Microsoft. The industry has been moving from the traditional model where organizations and companies keep their own databases isolated to a model where database storage and operations are outsourced to third parties. Outsourcing storage becomes increasingly popular, especially for those mobile devices with low local storage such as smart mobile phones and netbooks. Many new concepts, e.g., "software as a service", allow companies to provide useful functionality on remotely stored data. Subsequently, many new security and privacy issues of remote databases have arisen in such a "cloudy weather".

In this work, the author mainly investigates the cryptographic protocol based solutions to those security and privacy issues of remote databases. Historically, cryptographic protocols were used to achieve confidentiality, integrity and authenticity; however, modern complex cryptographic protocols go beyond those traditional goals, achieving a variety of other desired characteristics of computer-mediated collaboration. We design different cryptographic protocols for specific tasks in the following three scenarios.

In the first scenario, the remote server holds its own private database with sensitive records, and the clients will query the database. In one case, the client wants to fetch some elements from the database, but, for privacy, it is important to hide the clients' query patterns while preserving the server's interests as well, i.e., the client cannot obtain more than those queried elements. We propose our solution as adaptive *Oblivious Transfer* (OT) with fully simulation security. We propose a new efficient *Non-Interactive Zero-Knowledge* (NIZK) argument for correctness of shuffle, a result of independent importance. Sometimes, the client is not interested in the elements themselves but the statistical data of the database. We address this problem by constructing a generalized *Selective Private Function Evaluation* (SPFE) protocol where both the server's privacy and the client's privacy can be preserved.

In the second scenario, the clients outsource their database to some remote storage providers. The clients of such services do not always trust those storage providers, so the clients encrypt their outsourced data. It is necessary to have

a protocol that allows the client to update an element of the encrypted database without revealing to the semi-honest server which element was updated and to which value. The problem is addressed by so-called *PIR-writing* protocol, and we show two new communication-efficient *PIR-writing* protocols.

In the third scenario, the sensitive data is gathered from individuals and organizations, and multiple servers securely store the database in such a way that the adversarial server(s) cannot obtain any information about the database if certain security assumptions hold. Most secure *Multi-Party Computation* (MPC) systems can be used as oblivious databases where data is stored and processed in a secret-shared form. We design several round-efficient protocols for MPC, such as oblivious selection, filtering, sorting and shuffling — essential tools for *Privacy-Preserving Data-Mining* (PPDM).

ABBREVIATIONS

PIR	**P**rivate **I**nformation **R**etrieval	(c.f. Sect. 1.1.1)
PKS	**P**rivate **K**eyword **S**earch	(c.f. Sect. 1.1.1)
TTP	**T**rusted **T**hird **P**arty	(c.f. Sect. 1.1.1)
OT	**O**blivious **T**ransfer	(c.f. Sect. 1.1.1)
ZK	**Z**ero-**K**nowledge	(c.f. Sect. 1.1.1)
NIZK	**N**on-**I**nteractive **Z**ero-**K**nowledge	(c.f. Sect. 1.1.1)
SPFE	**S**elective **P**rivate **F**unction **E**valuation	(c.f. Sect. 1.1.1)
ORAM	**O**blivious **R**andom **A**ccess **M**achine	(c.f. Sect. 1.1.2)
PPDM	**P**rivacy **P**reserving **D**ata-**M**ining	(c.f. Sect. 1.1.3)
IC	**I**nference **C**ontrol	(c.f. Sect. 1.1.3)
MPC	**M**ulti-**P**arty **C**omputation	(c.f. Sect. 1.1.3)
DLIN	**D**ecisional **LIN**ear	(c.f. Sect. 2.1)
PKE	**P**ower **K**nowledge of **E**xponent	(c.f. Sect. 2.1)
PSDL	**P**ower **S**ymmetric **D**iscrete **L**ogarithm	(c.f. Sect. 2.1)
CPA	**C**hosen-**P**laintext **A**ttack	(c.f. Sect. 2.1)
LFCPA	**L**ength-**F**lexible **C**hosen-**P**laintext **A**ttack	(c.f. Sect. 2.2)
CRS	**C**ommon **R**eference **S**tring	(c.f. Sect. 2.2)
DAG	**D**irected **A**cyclic **G**raph	(c.f. Sect. 2.3)
BDD	**B**inary **D**ecision **D**iagram	(c.f. Sect. 2.3)
CPIR	**C**omputationally-**P**rivate **I**nformation **R**etrieval	(c.f. Sect. 2.3)
UC	**U**niversal **C**omposability	(c.f. Sect. 2.4)
OS	**O**blivious **S**torage	(c.f. Sect. 5.1)

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Recently, "Cloud computing" is being heavily promoted by market leaders such as Apple, Google and Microsoft. The industry has been moving from the traditional model where organizations and companies keep their own databases isolated to a model where database storages and operations are outsourced to third parties. Outsourcing storage becomes increasingly popular, especially for those mobile devices with low local storage, such as smart mobile phones and netbooks. Many new concepts, e.g., "software as a service", allow companies to provide useful functionality on remotely stored data. However, purely enforcing honest behavior through legislation, e.g., the broader European Union Directive on Data Protection [1], allows the database operators to disobey the law without detection. Subsequently, many new security and privacy issues have arisen in such a "cloudy weather". People are looking for solutions in mathematics, computer science and engineering aspects.

In this work, the author mainly investigates the cryptographic protocol based solutions to those security and privacy issues of remote databases. Historically, cryptographic protocols were used to achieve confidentiality, integrity and authenticity; however, modern complex cryptographic protocols go beyond those traditional goals, achieving a variety of other desired characteristics of computer-mediated collaboration. However, the most existing cryptographic protocols are less efficient, so we aim to construct more efficient cryptographic protocols for specific tasks in the following three scenarios.

1.1.1 Scenario One: Private Database Queries

In the first scenario, the remote server holds its own private database, and the clients will query the database. However, in many applications, where a client

remotely accesses a database, the client's query pattern is very sensitive. For example, if the database provider noticed that Alice has recently requested access to breast cancer related documents, then she might inadvertently reveal the information that either herself of someone closely related to her has breast cancer. Similarly, if a doctor from a psychiatric hospital queries a medical record of Bob, then there is a high chance that Bob is suffering from some psychiatric disease. In cryptography, these kinds of problems are addressed by cryptographic protocols such as *Private Information Retrieval* (PIR) and *Private Keyword Search* (PKS). In this work, we will study PIR. In a PIR protocol, a client can retrieve documents from a server in possession of a database without revealing which documents are retrieved. In this model, the data is public but centrally located, e.g., stock quotes. Since there is a trivial PIR, where the server sends the entire database to the client, only sublinear communication complexity PIR protocols are considered interesting. PIR was introduced by Chor *et al.* in 1995 [28], where they also showed that it is not possible to have a single-database PIR with sublinear communication complexity in the information-theoretic setting. In 1997, Kushilevitz and Ostrovsky [72] proposed a single-database PIR in the computational setting. Later, in 2004, Lipmaa [75] presented PIR protocols with log-squared communication complexity and another PIR with better communication complexity was shown by Gentry and Ramzan [46] in 2005.

In some cases, the server's private database contains sensitive records. For example, a server holds many movies and the clients can pay and watch them. For privacy protection, we need an approach that can hide the clients' query patterns while preserving the server's interests as well. One common solution is called *pseudonymization*, e.g., [27], which associates users with pseudonyms. A typical method is adding a layer between the client identities and actual database requests, and it requires a *Trusted Third Party* (TTP) to handle the pseudonym-to-identity mapping. Alternatively, we offer cryptographic solutions to this problem. In addition to PIR, the database's privacy is also required, i.e., the client is not able to obtain anything except the document(s) he/she retrieved. This problem is modeled by *Oblivious Transfer* (OT) in cryptography. The first closely-related protocol named "conjugate coding" was proposed by a physicist, Steven Weisner [94], in the 1970s. His protocol allows one to transmit two messages either but not both of which may be received, and the scheme relies on the quantum properties of transmitting individual photons from the sender to a polarizing filter on the receiver's side. Later, the idea of Weisner's protocol became the foundation of *quantum cryptography*. The notation of OT was formally introduced by Rabin [90] in 1981. We call his initial scheme Rabin's OT to avoid unnecessary confusion. In Rabin's OT, the server sends a document to a client with probability $\frac{1}{2}$, while the server remains oblivious as to whether the client successfully received

the document or not. 1-out-of-2 OT, denoted by OT_1^2, was later introduced by Even *et al.* [41] in 1982. In 1987, Crépeau [31] showed the equivalence between Rabin's OT and OT_1^2.

In this work, we investigate the so-called adaptive k-out-of-N OT that was introduced by Naor and Pinkas [84] in 1999. During an adaptive k-out-of-N OT, denoted by $\mathsf{OT}_{k\times 1}^N$, the client can adaptively fetch k documents from the server in possession of an N-document database such that the server gets no information about the client's selection and the client learns nothing more than those k documents. Adaptive OT is more useful than static OT in practice, and one classical application is oblivious search as mentioned in [84]. In an oblivious search protocol, Bob holds a database which Alice wants to search and obliviously determine whether the database contains a queried item. Bob sorts the database, and Alice uses binary search; therefore, Alice can perform oblivious search while Bob only reveals a limited proportion of the database, i.e., $\log N$. We will come back to this problem later with a more sophisticated solution. The security properties of most existing OT protocols are analyzed under a weaker security definition, the so-called "semi-simulation" (or "half-simulation") security. However, "semi-simulation" security is a weak security notion that is not always sufficient in practice. There exist attacks against semisimulatable protocols when they are used improperly [83]. Therefore, we focus on the construction of $\mathsf{OT}_{k\times 1}^N$ under stronger security definition known as "full-simulation" security, in which the security of the protocol is examined with respect to an ideal world where all parties only communicate with a *Trusted Third Party*.

We also construct a new efficient *Non-interactive Zero-knowledge* (NIZK) argument for correctness of shuffle. A shuffle permutes and re-encrypts a tuple of ciphertexts, and it is a well-known tool that helps to provide anonymity and obfuscation in applications like e-voting, mix-net, anonymous broadcast, etc. Verifiable shuffle has been studied for years. Abe [2] and Hoshino [4] constructed a 3-move proof for shuffle with size $\mathcal{O}(\kappa n \log n)$, where κ is the security parameter and n is the number of ciphertexts to be shuffled. In 2001, Neff [85] proposed a *zero-knowledge* (ZK) proof for shuffle of ElGamal ciphertexts. His shuffle is a 7-move ZK proof with size $\mathcal{O}(\kappa n)$. Later, Groth [56] generalized Neff's techniques. Furukawa and Sako [43] introduced a 3-move ZK argument for shuffle based on permutation matrix. Later, Furukawa [42] improved the ZK argument for shuffle in [43]. In 2008, Groth and Ishai [59] proposed the first sub-linear size ZK argument for shuffle. The only previous NIZK shuffle argument was presented by Groth and Lu [60]. In this work, we design a more efficient NIZK argument for shuffle. (c.f. Ch. 4, below.)

In some other cases, the clients are not interested in the information contained in individual elements of the database, and they require tools for private statistical

analysis of large databases. For example, Company A wants some statistical data from third-party databases for a marketing decision investigation, say the proportion of people that are within a certain range of age in a given region (e.g., ZIP code). It is obvious that the company does not want the database providers to know what the actual queries are for those queries may reveal crucial information about the company's future strategy. The problem is modeled as *Selective Private Function Evaluation* (SPFE), which is first introduced by Canetti *et al.* in 2001 [25]. In an SPFE protocol, the server (or multiple servers) holds a database $f = (f_0, \ldots, f_{n-1})$; the client wants to privately retrieve from the server the value $g(f_{x_1}, \ldots, f_{x_m})$ for some pre-defined m-argument function g and m indices $x_1, \ldots, x_m$ of the client's choice. After the protocol execution, the client can only learn the value of g on a selected sequence of m data items, while the server should learn nothing. Without loss of generality, g can be either known to both the client and the server or the client's private input. But in some real life scenarios, e.g., biometric authentication, SPFE is not enough. Suppose that employees are asked to be authenticated at the entrance of some building. The company has a pre-collected database of all its employees' fingerprints; the employee gives his/her ID and scans his/her fingerprint at the terminal. The terminal will run a protocol with the company's database to determine whether it is indeed the fingerprint of the corresponding employee. Due to human factors, device factors, and algorithm factors, it is extremely hard to have 100% accuracy in collecting and comparing two fingerprint samples. A well defined threshold should be given to obtain an optimal false negative rate and false positive rate. Since both the terminal and the database server can be corrupted, neither of them trust each other. Therefore, we would like to augment the definition of SPFE by allowing the client to have another private input y, i.e., the client will retrieve the value $g(f_{x_1}, \ldots, f_{x_m}, y)$. In our biometric authentication case, the client inputs an ID number: σ and a sample of fingerprint y, indicating that y is claimed to be the fingerprint of the employee with ID: σ. The server inputs the database of all fingerprints of the company's employees, denoted as $f = (f_0, \ldots, f_{n-1})$. The authentication protocol is essentially a generalized SPFE protocol $b \leftarrow g(f_\sigma, y)$, where g is the fingerprint matching algorithm that takes two fingerprints as input and outputs $b \in \{0, 1\}$, where '1' stands for acceptance and '0' stands for rejection. In this work, we propose several communication efficient generalized SPFE protocols and one concrete application for private similarity test.

1.1.2 Scenario Two: Outsourced Database

Considering the scenario of outsourced databases, the clients of such services do not always trust the storage provider to keep their privacy. Therefore, a client would like to only outsource an encrypted database that only he/she can decrypt.

On the other hand, the cloud is not just a disk in the sky, most applications require that the storage provider should allow clients to add, retrieve, modify and delete documents of their encrypted databases. This problem is addressed by so-called *PIR-writing* protocol (which is also known as private database modification [19]). In a *PIR-writing* protocol, the client updates one element of the encrypted database such that the semi-honest server does not get to know which element was updated and, of course, to which value. The trivial way is that the client downloads the entire database and updates the modified document(s). Then, the client re-encrypts the database and sends the new database back to the server. The main drawback is its linear communication costs to the size of the database, which is not interesting. The focus of our investigation is single-database *PIR-writing* protocol with sublinear communication complexity. The first non-trivial solution is proposed by Boneh *et al.* [19] in 2007. Their solution has communication complexity $\mathcal{O}(\sqrt{n})$ for modifying 1 bit of the database, where n is the size of the database. By repeating their protocol, one has a PIR-writing protocol with communication complexity $\mathcal{O}(\ell\sqrt{n})$ for modifying ℓ bits. Since the problem is close related to *Oblivious Random Access Machine* (ORAM), a brief overview of previous work is given in the corresponding chaper. (c.f. Ch. 5, below.) We also propose two more efficient *PIR-writing* protocols. The first one is based on Damgård-Jurik additively homomorphic public-key cryptosystem, and it has amortized poly-logarithmic communication for a limited number of updates. The second one is based on fully-homomorphic public-key cryptosystem, a much stronger primitive, but it achieves optimal logarithmic communication.

1.1.3 Scenario Three: Oblivious Database Manipulation

In the third scenario, the sensitive data was collected from individuals and organizations, and the data is stored in the cloud, performing numerous database analysis. *Privacy Preserving Data-Mining* (PPDM) has been thoroughly researched among different fields. A typical privacy-preserving data analysis application involves three types of entities: data donors, computing parties (referred to as miners), and clients. Data donors own sensitive data, miners gather the data, and clients want to query various statistical results. Some related topics, such as *Inference Control* (IC), address the problem that statistical results may reveal some sensitive individual information. For example, one may be entitled to query the average salary for female employees of a certain company, but there is exactly one female employee who works in this company. The query result directly tells the salary of the female employee. *Inference Control* provides the clients with access to a database for computing aggregate statistics about a collection of individuals while protecting the sensitive individuals' information in the database, such as query set restriction [5]. That is, the database monitors the query set of each

query, namely, a subset of records included in the computation of the response to the query, and limits the query set size, the overlap of query sets in successive queries by the same client, etc.

Alternatively, we are interested in giving cryptographic solutions to some other security issue of PPDM. Most of the secure *Multi-party Computation* (MPC) systems can be viewed as oblivious databases where data is stored and processed in a secret-shared form. The secure *Multi-party Computation* problem has been enjoying its popularity in cryptographic research community for decades. In 1982, the concept of MPC was initially introduced by Andrew C. Yao in [97] along with the famous classic millionaire problem: two millionaires want to know who is richer without revealing their actual wealth. An increasing number of practical MPC implementations, e.g., VIFF[34], Sharemind [16] and FairPlay [81, 14], witness a famous prediction made by Goldwasser [50] in 1997:

> "...the field of multi-party computations is today where public-key cryptography was ten years ago, namely an extremely powerful tool and rich theory whose real-life usage is at this time only beginning but will become in the future an integral part of our computing reality."

In a secure MPC scheme, n parties want to compute an agreed function of their inputs in a secure way such that it guarantees the correctness of the output and the privacy of the parties' inputs. Some parties might be dishonest or malicious. Let $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ to be the corresponding inputs and outputs of parties $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$, respectively. A (randomized) function is defined as $f(x_1, x_2, \ldots, x_n; r) = (y_1, y_2, \ldots, y_n)$, where r is a uniformly random value unknown to all parties. The MPC protocols aim to guarantee that party $\mathcal{P}_i$ will learn y_i after protocol execution, but it can obtain nothing more than that.

In such a setting, privacy issues can be modeled with the client-server model formalized by Damgård and Ishai [35]. In this model, data donors submit their data in a secret shared form to the miner nodes which later use share computing protocols to carry out the computations requested by the clients. Since data is secret shared, individual records are protected as long as the miners do not form non-tolerated coalitions, i.e.,, they follow the restrictions of share-computing protocols as a group. Clients and data donors are not trusted and can arbitrarily violate protocol specifications. Depending on the underlying primitives and protocols, the framework can tolerate either semi-honest or malicious corruption of miners.

Most MPC research only offers elementary operations, i.e., addition, multiplication, comparison and bit decomposition. On one hand, we have NAND (Negated AND) gates just with addition and multiplication, and NAND gates have

the functional completeness property such that all circuits can be represented by using only NAND gates. On the other hand, data manipulation in such databases can be slow and cumbersome without dedicated protocols for certain database operations. In this work, we investigate many essential tools in privacy-preserving data analysis, such as oblivious selection, filtering, sort and shuffle.

1.2 Contributions

This work is based on 5 published and 1 unpublished papers from the period of 2009 to 2011.

1. Lipmaa, H., Zhang, B.: Efficient Generalized Selective Private Function Evaluation with Applications in Biometric Authentication. In: Proceedings of the 5th Information Security and Cryptology, Inscrypt '09. LNCS, vol. 6151, pp. 154–163. Springer (2009).

   The author's main contribution was private similarity test part of the paper, and a better elaborated generic solution, which is constructed by the author (but is not included in the published version of the paper), shown in the corresponding chapter. (c.f. Ch 3, below.)

2. Zhang, B.: Simulatable Adaptive Oblivious Transfer With Statistical Receiver's Privacy. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 52–67. Springer (2011).

   The author is the only author of this paper and as such, both the constructions and the proofs are completed by the author himself.

3. Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Cryptology ePrint Archive, Report 2011/394 (2011), `http://eprint.iacr.org/2011/394`.

   The author came up with the initial version of the non-interactive zero-knowledge argument for correctness of shuffle. The argument was later significantly improved by his main supervisor (the co-author).

4. Lipmaa, H., Zhang, B.: Two New Efficient PIR-Writing Protocols. In: Proceedings of the 8th International Conference on Applied cryptography and network Security, ACNS '10. LNCS, vol. 6123, pp. 438–455. Springer (2010).

   The author is responsible for constructing the fully-homomorphic encryption based scheme that was proposed in the paper, together with corresponding proofs.

5. Zhang, B.: Generic Constant-Round Oblivious Sorting Algorithm for MPC. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 240–256. Springer (2011).

   The author is the only author of this paper and as such, both the constructions and the proofs are completed by the author himself.

6. Laur, S., Willemson, J., Zhang, B.: Round-efficient Oblivious Database Manipulation. In: Proceedings of the 14th Information Security Conference, ISC '11. LNCS, Springer (2011).

   The author came up with the initial idea and protocol constructions together with implementations, and they are generalized and refined by the other collaborative authors.

The copies of papers I–VI are included at the end of the thesis on pages 79 – 194.

1.3 Roadmap of the Thesis

The roadmap of the thesis is depicted in Figure 1.1. Chapter 2 provides necessary preliminaries and terminologies. Chapter 3 describes the formal definition and security of (generalized) SPFE protocol, and a better elaborated scheme was constructed in addition to the results of the corresponding included paper. Chapter 4 gives a formal definition and security of k-out-of-N adaptive OT and out result. Beside, a new NIZK argument for correctness of shuffle, a very important primitive, is proposed in the corresponding included paper. Chapter 5 gives the definition of PIR-writing protocols and lists our main results in the corresponding included paper. In addition, a brief overview of a close-related topic, *Oblivious RAM* (ORAM) is given. Chapter 6 provides a brief overview of using MPC for PPDM and describes our main results in the corresponding included papers. In Chapter 7, a short conclusion and future work is given.

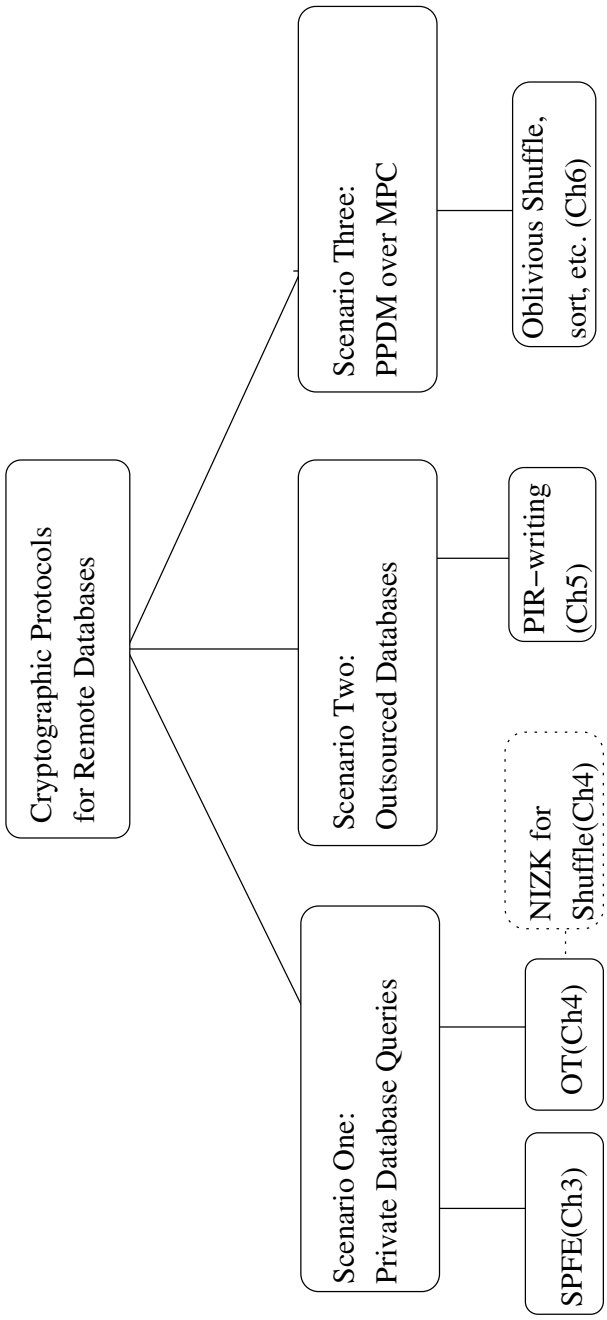


Figure 1.1: Roadmap of the Thesis

CHAPTER 2

PRELIMINARIES AND TERMINOLOGY

2.1 Notations

Throughout the thesis we use the following notations. We denote by $y \leftarrow P(x)$ the process of invoking the (presumably randomized) algorithm P on input x and assigning the result to y. Let $[n]$ denote the set $\{1, \ldots, n\}$. The term "p.p.t." is short for *probabilistic polynomial time Turing machine*. We overload G as probabilistic generator for different schemes and protocols except the bilinear group generator, which is denoted as Gen_{bp}. If the context is clear, we use the same notation G for different generators; otherwise, it depends on the specification. By $\mathbb{G} = \langle g \rangle$ we indicate that g is a generator of the cyclic group $\mathbb{G}$. Let S_n be the set of permutations from $[n]$ to $[n]$.

In the context of MPC, the message space is denoted by $\mathbb{K}$ if it is not specified. We denote the input array size by n and the number of parties by m. $\mathcal{P}_i$ stands for the i-th party. A shared value $\alpha \in \mathbb{K}$ is denoted by $[\![\alpha]\!]$ and the share held by $\mathcal{P}_i$ as $[\![\alpha]\!]_i$. In some protocols, R is the range of numbers to be sorted, and we assume that the numbers are from $[0, R]$. Let τ_{ad}, τ_{mul}, τ_{and}, τ_{or}, τ_{eq}, τ_{com} and τ_{bd} be the round complexity of addition, multiplication, unbounded fan-in AND, unbounded fan-in OR, equality check, comparison and bit-decomposition protocol of the underlying secure MPC, respectively.

Security Parameter. An adjustable security parameter κ is used in our cryptographic protocols. It is in unary representation 1^κ, which is a κ-bit string consisting of 1's. Therefore, the running time of the cryptographic algorithm can be specified as a function of the input size, κ. Sometime, the security parameter can be implicitly incorporated into other input parameters, such as group information.

Negligible Function.

Definition 1 (Negligible Function)**.** We say that a function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible if for every constant c there exists $N_c \in \mathbb{N}$ such that for all $x > N_c$ we have

$$\epsilon(x) < \frac{1}{x^c}.$$

Intuitively, a negligible function is asymptotically smaller than the inverse of any fixed polynomial. Examples of negligible functions include 2^{-n} and $n^{-\log\log n}$.

Statistical Distance. We use the following standard notion of statistical distance:

Definition 2 (Statistical Distance)**.** Let X, Y be random variables over the finite set U. Denote the distance between X and Y by

$$\mathsf{SD}(X,Y) \stackrel{def}{=} \max_{U' \subseteq U} \left| \Pr_{x \leftarrow X}[x \in U'] - \Pr_{y \leftarrow Y}[y \in U'] \right|.$$

Computational Indistinguishability. Let $\{X_\kappa\}_{\kappa\in\mathbb{N}}$ and $\{Y_\kappa\}_{\kappa\in\mathbb{N}}$ be ensembles of probability distributions where X_κ, Y_κ are probability distributions over $\{0,1\}^{\mathrm{poly}(\kappa)}$ for some polynomial function $\mathrm{poly}(\cdot)$. Let $\{X_\kappa\}_{\kappa\in\mathbb{N}} \stackrel{c}{\approx} \{Y_\kappa\}_{\kappa\in\mathbb{N}}$ denote that the computational indistinguishability of those distributions.

Definition 3 (Computational Indistinguishability)**.** We say that the ensembles $\{X_\kappa\}_{\kappa\in\mathbb{N}}$ and $\{Y_\kappa\}_{\kappa\in\mathbb{N}}$ are *computationally indistinguishable* if for all polynomial-time adversaries D and $\forall\kappa \in \mathbb{N}$:

$$|\Pr[s \leftarrow X_\kappa, D(s) = 1] - \Pr[s \leftarrow Y_\kappa, D(s) = 1]| \leq \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function.

(n, κ)-Nice Set.

Definition 4 ((n,κ)-Nice Set)**.** Let $n = \mathrm{poly}(\kappa)$. We say that $\Lambda = (\lambda_1, \ldots, \lambda_n) \subset \mathbb{Z}$ is an (n,κ)-*nice set*, if

$$0 < \lambda_1 < \cdots < \lambda_i < \cdots < \lambda_n = \mathrm{poly}(\kappa).$$

Bilinear Groups. Without loss of generality, we give the definition in asymmetric setting.

Definition 5 (Bilinear Group)**.** Let Gen_{bp} be a bilinear group generator such that $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa)$, where $\langle g_1 \rangle = \mathbb{G}_1, \langle g_2 \rangle = \mathbb{G}_2, \mathbb{G}_T$ are multiplicative cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is the bilinear map with the following properties:

1. Bilinearity: $\forall a, b \in \mathbb{Z}_p : e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.

2. Non-degeneracy: $\langle e(g_1, g_2) \rangle = \mathbb{G}_T$.

3. Efficiency: The map e must be efficiently computable.

A symmetric bilinear map $\hat{e}$ is a bilinear map on a single group. In practice, the symmetric bilinear maps can be constructed from asymmetric bilinear maps if there is an efficiently-computable isomorphism $\phi : \mathbb{G}_1 \to \mathbb{G}_2$ and $\hat{e}(x, y)$ is computed as $e(x, \phi(y))$. Asymmetric setting is more efficient in real implementation. We will use both settings in this work.

Decisional Linear Assumption (DLIN). The *Decisional Linear Problem* is introduced by Boneh, Boyen and Shacham [17]. Here, we give the definition in asymmetric setting.

Definition 6 (DLIN)**.** Let $t \in \{1, 2\}$. We say the DLIN holds for the bilinear group generator $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa)$ if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have

$$\begin{aligned} & Pr\left[gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); f, h \leftarrow \mathbb{G}_t^2; r, s \leftarrow \mathbb{Z}_p^2 : \mathcal{A}(gk, f, h, f^r, h^s, g_t^{r+s}) = 1\right] \\ \overset{c}{\approx} \; & Pr\left[gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); f, h \leftarrow \mathbb{G}_t^2; r, s, z \leftarrow \mathbb{Z}_p^3 : \mathcal{A}(gk, f, h, f^r, h^s, g_t^z) = 1\right]. \end{aligned}$$

Λ-Power Knowledge Of Exponent Assumption. Abe and Fehr [3] showed that it is impossible to construct a statistically *non-interactive zero-knowledge* argument for an $\mathbf{NP}$-complete language from a "black-box" security reduction to a standard cryptographic assumption, unless $\mathbf{NP} \subseteq \mathsf{P/poly}$. Another impossibility result was shown in [47]. We will base the soundness of our NIZK arguments (c.f. Ch. 4, below) on Λ-PKE, an explicit knowledge assumption. This assumption, proposed by Groth [58], is a generalization of the KE assumption of Damgård [32] and of the KEA3 assumption of Bellare and Palacio [13].

Let $t \in \{1, 2\}$. For two algorithms $\mathcal{A}$ and $\mathcal{X}_\mathcal{A}$, we denote $(y; z) \leftarrow (\mathcal{A}||\mathcal{X}_\mathcal{A})(x)$ for $\mathcal{A}$ on input x outputs y and $\mathcal{X}_\mathcal{A}$ on the same input, including the random tape of $\mathcal{A}$, outputs z. Let Λ be an (n, κ)-nice set for some $n = \mathrm{poly}(\kappa)$.

Definition 7 (Λ-PKE)**.** We say that the bilinear group generator Gen_{bp} is Λ-*PKE secure in group* $\mathbb{G}_t$ if for any non-uniform p.p.t. adversary $\mathcal{A}$ there exists a non-uniform p.p.t. extractor $\mathcal{X}_{\mathcal{A}}$ such that

$$Pr\left[\begin{array}{l} gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \\ (\alpha, x) \leftarrow \mathbb{Z}_p^2; \sigma = (g_t^\alpha, (g_t^{x^i}, g_t^{\alpha x^i})_{i\in\Lambda}); \\ (c, \hat{c}; (a_i)_{i\in\{0\}\cup\Lambda}) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(gk, \sigma) : \\ \hat{c} = c^\alpha \wedge c \neq \prod_{i\in\{0\}\cup\Lambda} g_t^{a_i x_i} \end{array}\right] < \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function.

Recall that gk contains $g_t = g_t^{x^0}$ but not $g_t^\alpha = g_t^{\alpha x^0}$. Groth [58] proved that the Λ-PKE assumption holds in the generic group model in the case $\lambda_i = i$; his proof can be straightforwardly modified to the general case. The special case where $\Lambda = \emptyset$, (i.e.,, the CRS contains only g_t^α, and the extractor returns a_0 such that $c = g_t^{a_0}$,) is similar to Damgård's original KE assumption [32], except that it is made in a bilinear group setting. We will write $\emptyset$-PKE as KE in the remaining part of the thesis.

Λ-Power Symmetric Discrete Logarithm Assumption. A version of the Λ-*Power Symmetric Discrete Logarithm Assumption (Λ-PSDL) in a non pairing-based group was defined in [65]. Lipmaa [77] proved that the Λ-PSDL assumption holds in the generic group model for any (n, κ)-nice set Λ given that $n = \mathrm{poly}(\kappa)$. We recap the Λ-PSDL that is used in [77] here.*

Definition 8 (Λ-PSDL)**.** We say that Λ-PSDL assumption holds for a bilinear group generator $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa)$, if for any non-uniform p.p.t. adversary $\mathcal{A}$, we have

$$Pr\left[\begin{array}{l} gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa); x \leftarrow \mathbb{Z}_p; \\ x' \leftarrow \mathcal{A}(gk, (g_1^{x^i}, g_2^{x^i})_{i\in\{0\}\cup\Lambda}) : x = x' \end{array}\right] < \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function.

2.2 Homomorphic Public-Key Encryption

In this work, most of the cryptographic protocols are based on homomorphic properties of public-key encryption schemes. Thanks to such useful properties, one can perform cryptographic operations on ciphertexts to manipulate the corresponding plaintexts. Denote $\mathsf{E}_{pk}(m; r)$ as encryption of plaintext m with randomizer r under public key pk. When the randomizer is not important in the context, we

describe encryption as $\mathsf{E}_{pk}(m;*)$ or $\mathsf{E}_{pk}(m)$ for short. Formally, we define homomorphic encryption as follow.

Definition 9 (Homomorphic Public-Key Encryption)**.** Let $(\mathbb{G}_1, \circledast), (\mathbb{G}_2, \odot)$ be two groups with a defined operation. $\mathcal{R}$ is the randomizer space. Given a security parameter κ, we say an encryption scheme $\mathsf{P} = (\mathsf{G}, \mathsf{E}, \mathsf{D})$, where G is a randomized key generation algorithm, E is a randomized encryption algorithm and D is a decryption algorithm

- $(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$
- $\mathsf{E}_{pk} : \mathbb{G}_1 \times \mathcal{R} \rightarrow \mathbb{G}_2$
- $\mathsf{D}_{sk} : \mathbb{G}_2 \rightarrow \mathbb{G}_1$

is homomorphic if $\mathsf{D}_{sk}(\mathsf{E}_{pk}(m_1; r_1) \odot \mathsf{E}_{pk}(m_2; r_2)) = m_1 \circledast m_2$.

Note that this definition is for general homomorphic cryptosystems with either multiplicatively or additively homomorphic properties. Usually, fully-homomorphic encryption schemes are over some finite ring with two group operations.

Length-Flexible Additively Homomorphic Encryption. Let $\mathsf{P}_{DJ} = (\mathsf{G}, \mathsf{E}, \mathsf{D})$ be a length-flexible additively-homomorphic public-key cryptosystem, e.g., [36], where G is a randomized key generation algorithm, E is a randomized encryption algorithm and D is a decryption algorithm. Here, both E and D receive an additional length parameter ℓ, so that $\mathsf{E}_{\mathsf{pk}}(\ell, \cdot)$ encrypts plaintexts from some set $\{0,1\}^{\leq \ell}$. In the case of the DJ01 cryptosystem from [36], for every integer $\ell > 0$, $\mathsf{E}_{\mathsf{pk}}(\ell, \cdot)$ is a valid plaintext of $\mathsf{E}_{\mathsf{pk}}(\lceil \ell/\kappa \rceil \cdot \kappa + \kappa, \cdot)$; therefore, one can multiple-encrypt messages as say in

$$c \leftarrow \mathsf{E}_{\mathsf{pk}}(\ell + 2\kappa, \mathsf{E}_{\mathsf{pk}}(\ell + \kappa, \mathsf{E}_{\mathsf{pk}}(\ell, m))) \ ,$$

and then recover m by multiple-decrypting,

$$m \leftarrow \mathsf{D}_{\mathsf{sk}}(\ell, \mathsf{D}_{\mathsf{sk}}(\ell + \kappa, \mathsf{D}_{\mathsf{sk}}(\ell + 2\kappa, c))) \ .$$

If the integer N is the public key of the DJ01 cryptosystem, then $2^\ell < N$. Additionally, in any length-flexible additively-homomorphic cryptosystem, $\mathsf{E}_{\mathsf{pk}}(\ell, m_1) \cdot \mathsf{E}_{\mathsf{pk}}(\ell, m_2) = \mathsf{E}_{\mathsf{pk}}(\ell, m_1 + m_2)$, where the addition is modulo the public key N. We will explicitly need the existence of a compression function C that, given pk, ℓ' and ℓ for $\ell' \geq \ell$, and $\mathsf{E}_{\mathsf{pk}}(\ell', m)$ for $m \in \{0,1\}^\ell$, returns $\mathsf{E}_{\mathsf{pk}}(\ell, m) \in \{0,1\}^{\lceil \ell/\kappa \rceil \cdot \kappa + \kappa}$. The compression function C can be simply implemented by modulo operation.

In the IND-CPA (*chosen-plaintext attack*) game, the challenger first generates a random $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{G}(1^\kappa)$ and sends pk to the adversary. The adversary chooses two messages (m_0, m_1) (such that $|m_0| = |m_1|$) and a length parameter ℓ, and sends them to the challenger. The challenger picks a random bit b and sends a ciphertext $\mathsf{E}_{\mathsf{pk}}(\ell, M_b)$ to the adversary. The adversary outputs a bit b' and wins if $b = b'$.

In the LFCPA (*length-flexible chosen-plaintext attack*) game [75], the challenger first generates a random $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{G}(1^\kappa)$ and sends pk to the adversary. The adversary chooses a polynomial number of message pairs (m_{j0}, m_{j1}) (such that $|m_{j0}| = |m_{j1}|$) and length parameters ℓ_j, and sends them to the challenger. The challenger picks a random bit b and sends all ciphertexts $\mathsf{E}_{\mathsf{pk}}(\ell_j, M_{jb})$ to the adversary. The adversary outputs a bit b' and wins if $b = b'$. Because of the existence of the compress function, LFCPA security follows from the IND-CPA security [76]. Thus, the DJ01 cryptosystem [36] is LFCPA-secure under the Decisional Composite Residuosity Assumption.

Lifted Knowledge BBS Encryption. BBS encryption was proposed by Boneh, Boyen and Shacham [17]. In original BBS encryption [17], the secret key is $(x, y) \in (\mathbb{Z}_p^*)^2$ and the public key is $(f = g^{1/x}, h = g^{1/y})$. To encrypt a message $m \in \mathbb{G}$ with randomizers $(s, t) \in \mathbb{Z}_p^2$, output $c = (c_1, c_2, c_3) = (f^s, h^t, m \cdot g^{s+t})$. To decryption a ciphertext c, output $m = c_3 \cdot c_1^{-x} \cdot c_2^{-y}$.

We will use a lifted "knowledge" version of this cryptosystem so that according to the KE assumption (similar to Damgård's original knowledge-of-exponent assumption), one can retrieve both the plaintext and the randomizer. Let $\mathsf{P}_{BBS} = (\mathsf{G}, \mathsf{E}, \mathsf{D})$ be the lifted knowledge BBS encryption. As always, G is a randomized key generation algorithm, E is a randomized encryption algorithm and D is a decryption algorithm. Let $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa)$.

Key Generation $\mathsf{G}(gk, 1^\kappa)$**:** Pick $(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3) \leftarrow \mathbb{Z}_p^3$, and denote $\tilde{g}_1 := g_1^{\tilde{\alpha}_3}, u := g_2^{\tilde{\alpha}_1}, v := g_2^{\tilde{\alpha}_2}, w := g_2^{\tilde{\alpha}_3}$. Pick secret keys: $sk := (x, y) \in (\mathbb{Z}_p^*)^2$, and set the public key as $pk := (gk, \tilde{g}_1, u, v, w, f, \tilde{f}, h, \tilde{h})$, where$f = (g_1^{1/x}, \tilde{f} = f^{\tilde{\alpha}_1}, h = g_1^{1/y}, \tilde{h} = f^{\tilde{\alpha}_2}$.

Encryption $\mathsf{E}_{pk}(m; s, t)$**:** To encrypt a message $m \in \mathbb{Z}_p$ with randomizer $(s, t) \in \mathbb{Z}_p^2$, output the ciphertext

$$C := (c_1, c_2, c_3, \tilde{c}_1, \tilde{c}_2, \tilde{c}_3) = (f^s, h^t, g_1^{m+s+t}, \tilde{f}^s, \tilde{h}^t, \tilde{g}_1^{m+s+t}).$$

Decryption $\mathsf{D}_{sk}(C)$**:** If $e(c_1, u) \neq e(\tilde{c}_1, g_2)$ or $e(c_2, v) \neq e(\tilde{c}_2, g_2)$ or $e(c_3, w) \neq e(\tilde{c}_3, g_2)$, return $\perp$. Otherwise, return the discrete logarithm of $g_1^m = c_3 \cdot c_1^{-x} \cdot c_2^{-y}$.

The lifted knowledge BBS cryptosystem is clearly additively homomorphic, since $\mathsf{E}_{pk}(m_1; s_1, t_2) \cdot \mathsf{E}_{pk}(m_2; s_2, t_2) = \mathsf{E}_{pk}(m_1 + m_2; s_1 + s_2, t_1 + t_2)$. One can also re-encrypt a ciphertext efficiently: if s_2 and t_2 are random, then $\mathsf{E}_{pk}(m; s_1, t_1) \cdot \mathsf{E}_{pk}(0; s_2, t_2) = \mathsf{E}_{pk}(m; s_1 + s_2, t_1 + t_2)$ is a random encryption of m. In our work [80], the lifted knowledge BBS cryptosystem is used. See our paper [80] for the details and security discussion.

Fully Homomorphic Encryption. In one of our protocols, fully homomorphic encryption scheme is used, where both multiplicative and additive circuit can be evaluated. A lot of variants of fully homomorphic encryption schemes have been proposed recently. Since we only used it as black box in our protocol, the details of the schemes are skipped here. Usually, those schemes are based on their simpler somewhat homomorphic versions that are homomorphic for small-depth circuits[1]. Take Gentry's scheme [44] as an example. As shown in [44], the somewhat homomorphic version of Gentry's cryptosystem is sufficient to homomorphically evaluate its own decryption circuit augmented with basic Boolean operations. Hence, one can strengthen the somewhat homomorphic version with a *bootstrapping* step. Assume that the plaintexts have been encrypted by using some public key pk_1. Now, just before the circuit depth has reached the level where decryption becomes incorrect, one encrypts the ciphertexts $\mathsf{E}_{\mathsf{pk}_1}(\cdot)$ by using a different public key pk_2, and then homomorphically decrypts the results, obtaining new encryptions of the same plaintexts but under the new key pk_2 and with decreased noise. After that, one can continue homomorphically executing another few levels of the circuit, until one needs to bootstrap again. According to recent eprint by Gentry [45], one can construct fully-homomorphic encryption without bootstrapping.

2.3 Cryptocomputing and Branching Program Evaluation

Cryptocomputing. Let $m, \ell \in \mathbb{N}$ be public parameters, and let $\mathcal{F}$ a class of functions $\{0,1\}^m \to \{0,1\}^\ell$. In a cryptocomputing protocol for $\mathcal{F}$ between a client and a server, the client has an input $x \in \{0,1\}^m$ and the server has an input $f \in \mathcal{F}$. The client obtains $f(x)$. Every cryptocomputing protocol $\Gamma = (\mathsf{G}, \mathsf{Q}, \mathsf{R}, \mathsf{A})$ has two messages where the client generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{G}(1^\kappa)$. The client sends $\mathsf{pk}, q \leftarrow \mathsf{Q}(\mathsf{pk}, \ell, x)$ to the server, the server replies with $r \leftarrow \mathsf{R}(\mathsf{pk}, \ell, f, q)$, and then finally the stateful client recovers $f(x)$ by computing

[1] Typically, we only count the multiplicative depth of an arithmetic circuit, for the additive circuit does not increase the "error" so much, comparing to the multiplicative circuit.

$\mathsf{A}(\mathsf{sk}, \ell, x, r)$. Here, G, Q, R and A are (probabilistic) polynomial-time algorithms. When the context is clear, we hide G and pk, sk for simplicity.

Branching Program.

Definition 10 (Branching Program)**.** A (deterministic) branching program over the variables $x := (x_1, \ldots, x_n)$ with input domain I and output domain O is defined by a tuple $(G = (V, E), v_s, T, \psi_V, \psi_E)$ where:

- G is a *directed acyclic graph* (DAG).
- v_s is an initial node of in-degree 0, and $\forall v \in V \setminus \{v_s\}$ are reachable from v_s.
- $T \subseteq V$ is a set of sink nodes (or terminal nodes) whose out-degrees are 0.
- $\psi_V : V \to [n] \cup O$ is a node-labeling function assigning a variable index from $[n]$ to each non-terminal node $w \in V \setminus T$ and an output value to each sink node $u \in T$.
- $\psi_E : E \to I$ is an edge labeling function such that every edge is mapped to a non-empty partition of input I.

In our work, we assume that branching programs have binary inputs, namely $I = \{0, 1\}$. Branching programs are also known as *Binary Decision Diagram* (BDD) [93]. The output $\mathsf{BDD}(x)$ of a binary decision diagram BDD on an input $x \in I^n$ is naturally defined by following the path induced by x from v_s to a sink node $v_t \in T$, where the successor of node v is the unique node v' such that $x_{\psi_V(v)} \in \psi_E(v, v')$. The output is the value $\psi_V(v_t)$ labeling the sink node reached by the path. The size of BDD, denoted as $\mathsf{size}(\mathsf{BDD})$, is $|E|$, which is also $\mathcal{O}(|V|)$ for binary inputs. The height of a node $v \in V$, denoting as $\mathsf{height}(v)$, is the length of the longest path from v to a sink node $v_t \in T$. The depth of BDD, denoted as $\mathsf{len}(\mathsf{BDD})$, is defined as $\mathsf{height}(v_s)$. Cobham [29] showed that any language in L/poly can be computed by polynomial-size branching program. In a multi-terminal BDD, the DAG has more than one initial nodes, and the value of each initial node should be evaluated respectively. Figure 2.1 shows an example of BDD for Boolean function $f(a, b, c, d) = (a \wedge b \wedge c) \vee d$.

Computationally-Private Information Retrieval. A two-message 1-out-of-n *computationally-private information retrieval* protocol, denoted as $(1, n)$-CPIR, is a special type of cryptocomputing protocol. In a $(1, n)$-CPIR protocol for ℓ-bit strings, the client has an index $x \in \{0, \ldots, n-1\}$ and the server has a database $f = (f_0, \ldots, f_{n-1})$ with $f_i \in \{0, 1\}^\ell$. The client obtains f_x. An $(1, n)$-CPIR

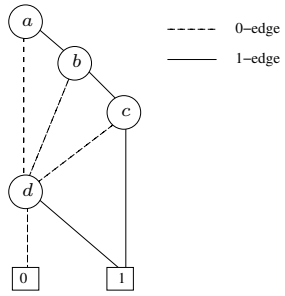


Figure 2.1: BDD for Boolean function $f(a, b, c, d) = (a \wedge b \wedge c) \vee d$

protocol $\Gamma_{cpir} = (\mathsf{G}, \mathsf{Q}, \mathsf{R}, \mathsf{A}, \mathsf{C})$ is *BDD-friendly* if it satisfies the next four assumptions:

1. Γ_{cpir} is a two-message protocol. The client computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{G}(1^\kappa)$ and a query $q \leftarrow \mathsf{Q}(\mathsf{pk}, \ell, x)$, and the client sends pk, q to the server. The server replies $r \leftarrow \mathsf{R}(\mathsf{pk}, \ell, f, q)$, such that the stateful client can recover f_x by computing $f_x \leftarrow \mathsf{A}(\mathsf{sk}, \ell, x, r)$.

2. Γ_{cpir} is uniform in ℓ; that is, it can be easily modified to work on other values of ℓ.

3. $|\mathsf{Q}(\mathsf{pk}, \ell, \cdot)|, |\mathsf{R}(\mathsf{pk}, \ell, \cdot, \cdot)| \leq \ell + \Theta(\kappa)$ (with possibly $\mathsf{Q}(\mathsf{pk}, \ell, \cdot)$ being even shorter).

4. The compress function C maps $\mathsf{Q}(\mathsf{pk}, \ell', x)$ to $\mathsf{Q}(\mathsf{pk}, \ell, x)$ for any $\ell' \geq \ell$ and any x.

Here G, Q, R, A and C are (probabilistic) polynomial-time algorithms. The only known BDD-friendly $(1, 2)$-CPIR was proposed by Lipmaa in [75], see [76] for a compact description. In Lipmaa's $(1, 2)$-CPIR protocol, $\mathsf{Q}(\mathsf{pk}, \ell, x)$ consists of an additively homomorphic encryption of x under public key pk.

Any $(1, n)$-CPIR protocol Γ_{cpir} must be client-private, i.e.,, IND-CPA secure. Lipmaa's $(1, 2)$-CPIR protocol [75], when based on the DJ01 cryptosystem [36], is IND-CPA secure and thus LFCPA-secure (which is defined in the same way as LFCPA-security for public-key cryptosystems) under the Decisional Composite Residuosity Assumption.

PrivateBDD Protocol. In [64], Ishai and Paskin proposed a new cryptocomputing method (PrivateBDD) that uses a BDD-representation of the target function in conjunction with a communication-efficient strong oblivious transfer. In [76],

the authors noted that the strong oblivious transfer protocol can be replaced by a BDD-friendly $(1,2)$-CPIR protocol. In addition, the authors of [76] also improved the concrete efficiency of the PrivateBDD protocol. We now briefly recall the main properties of PrivateBDD as instantiated by Lipmaa's $(1,2)$-CPIR from [75]. See [76] for the full details of the PrivateBDD protocol.

Theorem 1. *Assume that the Decisional Composite Residuosity Assumption is true. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^m \to \{0,1\}^\ell$, and for any $f \in \mathcal{F}$ let P_f be some (multi-terminal) BDD with ℓ-bit sink labels that computes f. Let $\mathsf{len}(P_\mathcal{F}) := \max_{f\in\mathcal{F}} \mathsf{len}(P_f)$. Then $\mathcal{F}$ has a IND-CPA secure cryptocomputing protocol with communication upperbounded by $\kappa + m \cdot (\ell + (\mathsf{len}(P_\mathcal{F}) + 2) \cdot \kappa)$, and server's online computation dominated by $\mathsf{size}(P_f)$ public-key operations.*

Briefly, the client's inputs to the PrivateBDD (when instantiated by Lipmaa's $(1,2)$-CPIR from [75]) are encrypted bitwise by using a length-flexible additively homomorphic public-key cryptosystem like DJ01 [36]. Moreover, let v be any internal node of the BDD such that $\mathsf{height}(v) > 0$. ($\mathsf{height}(v)$ is the longest path between v and any sink node). Let v_0 and v_1 be the successors of v by the 0-edge and 1-edge, respectively. Then v's value, denoted by $val[v]$, as recursively computed by the PrivateBDD protocol is

$$\mathsf{R}(\mathsf{pk}, \ell + (\mathsf{height}(v) - 1)\kappa, (val[v_0], val[v_1]), \mathsf{Q}(\mathsf{pk}, \ell + (\mathsf{height}(v) - 1)\kappa, x_j)) \ ,$$

where x_j is v's label, and $val[v_i]$ is the already known value of the node v_i. Moreover, sink values are equal to their labels. Therefore, $val[v]$ is equal to an encryption of $val[v_{x_j}]$. Inductively, $val[v]$ is equal to an $\mathsf{height}(v)$-times encryption of some sink value, and $|val[v]| \approx (\mathsf{height}(v) + 1)\kappa$. In particular, the server's message in the PrivateBDD protocol is equal to a $\mathsf{len}(P_f)$-times encryption of some sink value, and this sink value by itself is the output of the PrivateBDD protocol, where P_f is the corresponding BDD for function f. See [76] for more details.

Security of Cryptocomputing Protocols. We recap the security definition used in [64], which is so-called semi-simulatable or half-simulatable security.

Definition 11 (Representation Model)**.** A representation model is a polynomial-time computable function $U : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, where $U(P,x)$ is referred to as the value returned by a "program" P on the input x. When U is understood from the context, we use $P(x)$ to denote $U(P,x)$. We say that a function $f : \{0,1\}^* \to \{0,1\}^*$ can be implemented in a representation model U if there exists an infinite sequence $(P_0, P_1, \ldots)$, referred to as an implementation of f in U, such that $f(x) = U(P_{|x|}, x)$ for every $x \in \{0,1\}^*$.

Definition 12 (Client's Privacy)**.** Let $\Gamma = (\mathsf{G}, \mathsf{Q}, \mathsf{R}, \mathsf{A})$ be a cryptocomputing protocol. We say that Γ satisfies the client's privacy requirement if for all p.p.t. adversary $\mathcal{A}$ we have

$$\left| Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{G}(1^\kappa); (x_0, x_1 \in \{0,1\}^*, \ell \in \mathbb{N}) \leftarrow \mathcal{A}(\mathsf{pk}) \\ s.t. |x_0| = |x_1| \leq \ell; b \leftarrow \{0,1\}; q \leftarrow \mathsf{Q}(\mathsf{pk}, \ell, x_b); \\ b' \leftarrow \mathcal{A}(\mathsf{pk}, q) : b = b' \end{array} \right] - \frac{1}{2} \right| < \epsilon(\kappa)$$

Note that the client's privacy in (two-move) PrivateBDD protocol is analogous to the definition of IND-CPA security of the underlying cryptosystems.

Definition 13 (Server's Privacy: Semi-honest Model)**.** Let $\Gamma = (\mathsf{G}, \mathsf{Q}, \mathsf{R}, \mathsf{A})$ be a cryptocomputing protocol for evaluating programs from a representation model U on encrypted data. We say that Γ has statistical server privacy in the semi-honest model if there exists a p.p.t.-simulator Sim such that the following holds. For every security parameter κ, input $x \in \{0,1\}^*$ output $y \in \{0,1\}^\ell$, pair (pk, q) that can be generated by G, Q on inputs κ, x, and program $P \in \{0,1\}^*$, we have

$$\mathsf{SD}(\mathsf{R}(\mathsf{pk}, \ell, P, q), \mathsf{Sim}(\mathsf{pk}, 1^{|x|}, \ell, U(P,x), 1^{|P|}) \leq \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function. The case of perfect server's privacy is defined similarly, except that $\epsilon(\kappa) = 0$ and Sim are allowed to run in expected polynomial time. In the case of computational server's privacy, Sim should satisfy the following requirement. For every polynomial-time D, for any $\kappa, x, \ell, \mathsf{pk}, q$ and P we have

$$\Pr[D(\mathsf{R}(\mathsf{pk}, \ell, P, q)) = 1] - \Pr[D(\mathsf{Sim}(\mathsf{pk}, 1^{|x|}, \ell, U(P,x), 1^{|P|})) = 1] \leq \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function.

2.4 Security Definitions for MPC

Adversarial model. For clarity, we consider only the static corruption model where the adversary specifies parties to be corrupted before the protocol starts, although most protocols can resist more advanced corruption models. Although the list of tolerated adversarial coalitions can be arbitrary, share computing systems can achieve information theoretical security only if the condition Q2 is satisfied in the semi-honest model and the condition Q3 is satisfied in the malicious model [63]. Recall that the condition Q2 means that any union of two tolerated adversarial coalitions is not sufficient to corrupt all parties and the condition Q3 means that any union of three tolerated adversarial sets is not sufficient. In the case of threshold corruption, the conditions Q2 and Q3 imply that the number corrupted parties is strictly below $\frac{m}{2}$ and $\frac{m}{3}$, respectively.

Universal Composability. As formal security proofs are rather technical, security proofs are often reduced to the security properties of sub-protocols. More specifically, one can deduce the security of a compound protocol without delving into details only if all sub-protocols are *universally composable* (UC). Although the formal definition of universal composability is rather complex, the intuition behind it is simple. A protocol ψ is UC-secure if there exists no p.p.t.-environment $\mathcal{Z}$ that can distinguish whether it is interacting with the real world adversary $\mathcal{A}$ and parties running protocol ψ or with the ideal adversary $\mathcal{A}'$ and dummy parties interacting with an ideal functionality $\mathcal{F}_\psi$.

Definition 14. We say that a protocol ψ UC-realize $\mathcal{F}_\psi$ if for any adversary $\mathcal{A}$, there exists a $\mathcal{A}'$ such that for all environments $\mathcal{Z}$ we have

$$\mathbf{Ideal}_{\mathcal{F}_\psi,\mathcal{A}',\mathcal{Z}} \overset{c}{\approx} \mathbf{Real}_{\psi,\mathcal{A},\mathcal{Z}}.$$

As a result, a compound protocol consisting of several instances of $\mathcal{F}_\psi$ preserves security if we replace $\mathcal{F}_\psi$ by ψ. It means that we combine universally composable sub-protocols without any usage restrictions, e.g.,, execute them in parallel. We refer to the standard treatments [24, 87] for further details.

2.5 Zero-knowledge Proofs

(Interactive) Zero-knowledge Proof and Σ-Protocol. We follow the definitions described in [30, 12]. A pair of interactive algorithms $(\mathcal{P}, \mathcal{V})$, called a prover and a verifier is a proof of knowledge for a relation $\mathcal{R} = \{(\alpha, \beta)\} \subseteq \{0,1\}^* \times \{0,1\}^*$ with knowledge error $\kappa \in [0,1]$ if for all $(\alpha, \beta) \in \mathcal{R}$, $\mathcal{V}(\alpha)$ accepts a conversation with $\mathcal{P}(\beta)$ with probability 1, and there exists a polynomial time knowledge extractor $\mathcal{E}$ such that if a cheating prover $\hat{\mathcal{P}}$ has probability ε of convincing $\mathcal{V}$ to accept α, then $\mathcal{E}$ outputs a witness β for α with probability $\varepsilon - \kappa$ via rewindable black-box access to $\hat{\mathcal{P}}$.

A proof system $(\mathcal{P}, \mathcal{V})$ is *computational honest-verifier zero-knowledge* if there exists a p.p.t.-simulator $\mathcal{S}$ such that for any $(\alpha, \beta) \in \mathcal{R}$, the outputs of $\mathcal{V}(\alpha)$ after interacting with $\mathcal{P}(\beta)$ or with $\mathcal{S}(\alpha)$ are computationally indistinguishable. It is possible to transform an honest-verifier zero-knowledge proof system to a general zero-knowledge one, e.g., [30]. Therefore, our ZK argument will be honest-verifier zero-knowledge throughout the paper.

A Σ-Protocol for language $\mathcal{L}$ is a proof system $(\mathcal{P}, \mathcal{V})$ where the conversation is a tuple (α, β, γ), where $\mathcal{P}$ outputs α and $\mathcal{V}$ gives a random challenge β, and then $\mathcal{P}$ replies γ. $\mathcal{V}$ accepts if $\phi(x, \alpha, \beta, \gamma) = 1$, where ϕ is a predicate function. A Σ-protocol must satisfy three security properties: correctness, special soundness and special honest-verifier zero knowledge. A Σ-protocol is correct

when an honest prover convinces an honest verifier with probability $1 - k^{-\omega(1)}$. A Σ-protocol has the special soundness property when from two accepted views (α, β, γ) and $(\alpha, \beta', \gamma')$, where $\beta \neq \beta'$, one can efficiently recover a witness w such that $(x, w) \in \mathcal{R}$. A Σ-protocol has the special honest-verifier zero-knowledge property if there exists a p.p.t. simulator $\mathcal{S}$ that can output a tuple $(\alpha^*, \beta^*, \gamma^*)$ that will be accepted and such the distribution of $(\alpha^*, \beta^*, \gamma^*)$ is computationally indistinguishable from the distribution of accepted views between an honest prover and an honest verifier.

Non-Interactive Zero-knowledge Argument for Group-Specific Languages. A new *non-interactive zero-knowledge* (NIZK) proof technique was introduced in [61], and subsequently, many efficient NIZK arguments are constructed, e.g., [60, 57]. Let $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{Gen}_{bp}(1^\kappa)$ be a bilinear group. Let $\mathcal{R} = \{gk; S, W\}$ be an efficiently computable (group-specific) binary relation such that $|W| = \mathrm{poly}(|S|)$, where S is a statement and W is a witness. Let $\mathcal{L} = \{(gk; S) : \exists W, (gk; S, W) \in \mathcal{R}\}$ be a (group-specific) $\mathbf{NP}$-language. For example, multiplication, (multi-)exponent and shuffle are naturally group-specific languages, for one proves relations between elements of the same bilinear group.

A *non-interactive argument* for $\mathcal{R}$ consists of the following probabilistic polynomial-time algorithms: a group setup algorithm Gen_{bp}, a CRS generation algorithm G, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. For $gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa), \mathsf{crs} \leftarrow \mathsf{G}(gk), \mathcal{P}(gk, \mathsf{crs}; S, W)$ outputs an argument ψ. The verifier $b \leftarrow \mathcal{V}(gk, \mathsf{crs}; S, \psi)$, where $b \in \{0, 1\}$ such that 1 stands for acceptance and 0 stands for rejection.

Definition 15 (Perfect Completeness)**.** We say that a non-interactive argument $(\mathsf{Gen}_{bp}, \mathsf{G}, \mathcal{P}, \mathcal{V})$ is perfect complete, if for all non-uniform adversaries $\mathcal{A}$ we have:

$$Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \mathsf{crs} \leftarrow \mathsf{G}(gk); (S, W) \leftarrow \mathcal{A}(gk, \mathsf{crs}); \\ \psi \leftarrow \mathcal{P}(gk, \mathsf{crs}; S, W) : \\ (gk; S, W) \notin \mathcal{R} \vee \mathcal{V}(gk, \mathsf{crs}; S, \psi) = 1 \end{array}\right] = 1$$

Definition 16 (Computational Soundness)**.** We say that a non-interactive argument $(\mathsf{Gen}_{bp}, \mathsf{G}, \mathcal{P}, \mathcal{V})$ is computationally sound, if for all non-uniform adversaries $\mathcal{A}$ we have:

$$Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \mathsf{crs} \leftarrow \mathsf{G}(gk); (S, \psi) \leftarrow \mathcal{A}(gk, \mathsf{crs}) : \\ (gk; S) \notin \mathcal{L} \wedge \mathcal{V}(gk, \mathsf{crs}; S, \psi) = 1 \end{array}\right] < \epsilon(\kappa),$$

where $\epsilon(\cdot)$ is some negligible function.

Definition 17 (Perfectly Witness-indistinguishability)**.** We say that a non-interactive argument $(\mathsf{Gen}_{bp}, \mathsf{G}, \mathcal{P}, \mathcal{V})$ is perfectly witness-indistinguishable, if for all non-uniform adversaries $\mathcal{A}$ we have:

$$\begin{aligned} & Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \mathsf{crs} \leftarrow \mathsf{G}(gk); (S, W_0, W_1) \leftarrow \mathcal{A}(gk, \mathsf{crs}); \\ \psi \leftarrow \mathcal{P}(gk, \mathsf{crs}; S, W_0) : \mathcal{A}(\psi) = 1 \end{array}\right] \\ = \quad & Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \mathsf{crs} \leftarrow \mathsf{G}(gk); (S, W_0, W_1) \leftarrow \mathcal{A}(gk, \mathsf{crs}); \\ \psi \leftarrow \mathcal{P}(gk, \mathsf{crs}; S, W_1) : \mathcal{A}(\psi) = 1 \end{array}\right], \end{aligned}$$

where we require $(gk; S, W_0), (gk; S, W_1) \in \mathcal{R}$.

Definition 18 (Perfect Zero-knowledge)**.** We say that a non-interactive argument $(\mathsf{Gen}_{bp}, \mathsf{G}, \mathcal{P}, \mathcal{V})$ is perfectly zero-knowledge, if there exists a polynomial-time simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, such that for all stateful interactive non-uniform adversaries $\mathcal{A}$,

$$\begin{aligned} & Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); \mathsf{crs} \leftarrow \mathsf{G}(gk); (S, W) \leftarrow \mathcal{A}(gk, \mathsf{crs}); \\ \psi \leftarrow \mathcal{P}(gk, \mathsf{crs}; S, W) : (gk; S, W) \notin \mathcal{R} \wedge \mathcal{A}(\psi) = 1 \end{array}\right] \\ = \quad & Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Gen}_{bp}(1^\kappa); (\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(gk); (S, W) \leftarrow \mathcal{A}(gk, \mathsf{crs}); \\ \psi \leftarrow \mathcal{S}_2(gk, \mathsf{crs}; S, \tau) : (gk; S, W) \notin \mathcal{R} \wedge \mathcal{A}(\psi) = 1 \end{array}\right], \end{aligned}$$

where τ is the *simulation trapdoor*.

CHAPTER 3

GENERALIZED SELECTIVE PRIVATE FUNCTION EVALUATION

In this chapter, we give definition of generalized *selective private function evaluation*. In addition to the results presented by the author and his supervisor in the paper "Efficient Generalized Selective Private Function Evaluation with Applications in Biometric Authentication" [78], we give a new two-move protocol that improves on the two-move protocol in the paper [78].

3.1 Selective Private Function Evaluation

Selective Private Function Evaluation (SPFE) was first introduced by Canetti *et al.* in 2001 [25]. In an SPFE protocol, the server (or multiple servers) hold(s) a database $f := (f_0, \ldots, f_{n-1})$; the client wants to privately retrieve from the server(s) the value $g(f_{x_1}, \ldots, f_{x_m})$ for some m-argument function g and m indices $x_1, \ldots, x_m$ of the client's choice. After the protocol execution, the client only can learn the value of g on a selected sequence of m data items, while the server should learn nothing. Without loss of generality, g can be either known to both the client and the server or the client's private input. In our work, we focus on the single-server SPFE protocols where the server and the client have a common public agreed g and m; The solution to private g can be obtained by defining g as a "universal function" and the client has additional input to specify the actual function to be evaluated.

Generalized Selective Private Function Evaluation. We would like to augment the definition of SPFE by allowing the client to have another private input y, i.e., the client will retrieve the value $g(f_{x_1}, \ldots, f_{x_m}, y)$, and we give the formal definition as follows.

Definition 19. Let $k, n, m \in \mathbb{N}$, let D, Y be some finite domain. In a (k, n, m)-SPFE protocol, there are $k+1$ parties, the client $\mathbf{C}$ and k servers $\mathbf{S}_1, \ldots, \mathbf{S}_k$. The servers have a common input $f \in D^n$ representing the data, and the client has a deterministic function $g : D^m \times Y \to D$, where $m < n$, a list $I \in [n]^m$ of m indices and an additional input $y \in Y$. After the protocol execution, the servers output $\perp$, and the client outputs $g(f_I, y)$, where $f_I \stackrel{def}{=} (f_{x_1}, \ldots, f_{x_m})$.

The single-server scenario is a special case of generalized SPFE, denoting as (n, m)-SPFE, which is short for $(1, n, m)$-SPFE.

Security. Let $\kappa, t \in \mathbb{N}$. All parties are assumed to be polynomial in security parameter κ, which implies that the database size, $n < \mathrm{poly}(\kappa)$. For the sake of uniformity, we formulate our security requirements only against polynomial-time adversaries. Although, in multiple servers scenario, the security can hold even against computationally unbounded adversaries. We have the following three requirements (informally).

1. Correctness: If both the client and the servers follow the protocol, then the client's output will be the correct value of $g(f_I, y)$.

2. Client's privacy: there exist no p.p.t. adversary that controls up to t servers that can learn anything from the protocol execution more than pre-defined information, e.g., the function g and the number of indices m, even if the corrupted servers deviate from the protocol in an arbitrary way.

3. Server's secrecy: there exists no p.p.t. adversarial client that can learn anything from the protocol execution more than pre-defined information about the data, i.e., $g(f_I, y)$, even if the client deviate from the protocol in an arbitrary way.

In terms of two-move generalized SPFE protocol, the security definition is similar to the security definition of crypto-computing protocol mentioned in preliminary. We use the convention of many previous papers to only require semi-simulatable privacy in the malicious model. The client's privacy is guaranteed in the sense of indistinguishability of any two client's input, and the server's privacy is guaranteed in the sense of simulatability. (cf. sect. 2.3.)

Two Generalized SPFE Protocols. We give an overview of two efficient generalized SPFE protocol based on PrivateBDD in the included paper [78]. The first protocol has two moves, but it only works for constant m, and it is especially efficient in the case $m = 1$. In this protocol, the server computes a database of the answers $g(f_I, y)$ for all possible $f_I \in [n]^m$, using the PrivateBDD protocol. Then

the client and the server run an efficient two-message $(1, \binom{n}{m})$-CPIR protocol on this database so that the client retrieves $g(f_I, y)$. An improved two-move protocol is given in Sect. 3.2.

The second protocol works for any value of m. It consists of an input selection protocol, after which the client obtains values $f_{x_1} \oplus r_1, \ldots, f_{x_m} \oplus r_m$ for random strings r_j that are chosen by the server. That is, the client and the server secret-share the values f_{x_j}. It is then followed by a PrivateBDD protocol, where client's inputs are $f_{x_j} \oplus r_j$ and y, server's inputs are g and r_j, and client's output is $g(f_I, y)$. Due to the homomorphic properties of the underlying $(1, 2)$-CPIR protocol by Lipmaa [75], it is straightforward to implement PrivateBDD on secret-shared inputs f_{x_j} based on a PrivateBDD protocol that uses non-shared input. The resulting generalized SPFE protocol has 4 moves, and requires to compute $(1, n)$-CPIR m times, and then to execute PrivateBDD for g. This protocol has sublinear communication whenever $\mathsf{len}(P_g)$ is sublinear in n, where P_g is the BDD for function g.

3.2 New Two-move Generalized SPFE Protocol

We propose a new two-move generalized SPFE protocol that is more efficient than the two-move generalized SPFE protocol proposed in the paper [78] in terms of computation complexity. The idea is to disassemble CPIR protocols and integrate them into the BDD of the statistical function g. Disassemble the $(1, n)$-CPIR for document with length ℓ as ℓ distinct CPIRs h_j that take input as index $x \in \{0,1\}^{\lceil \log n \rceil}$ and output the j-th bit of document f_x, denoted as $f_{x,j}$. h_j can be regarded as a boolean function that takes as input an $\lceil \log n \rceil$-bit index and outputs a single bit. To fetch an ℓ-bit document, the client sends bit-wise encryption of x, and the server evaluates ℓ functions $f_{x,j} \leftarrow h_j(x)$ in a parallel, for $1 \leq j \leq \ell$. Figure 3.1 shows an example of $(1, 8)$-CPIR. The left side is the original disassembled CPIR, and the right side is the corresponding BDD for its boolean function $f(x_1, x_2, x_3)$.

Note that for any boolean function $f : \{0,1\}^\lambda \to \{0,1\}$, the size of its BDD is $\mathsf{size}(P_f) \leq (1 + o(1))2^\lambda/\lambda$ [21]. We employ the PrivateBDD-based $(1, n)$-CPIR that was proposed by Lipmaa [76]. Let P_{h_j} be the BDD for function h_j. Define $\mathsf{size}(P_H) \stackrel{def}{=} \max_j \mathsf{size}(P_{h_j})$ and $\mathsf{len}(P_H) \stackrel{def}{=} \max_j \mathsf{len}(P_{h_j})$. The server's online computational complexity is $\Theta(\ell \cdot \mathsf{size}(P_H)) = \Theta(\ell \cdot \frac{n}{\log n})$ public-key operations, and communication complexity is $\Theta(\log n \cdot \ell + \log n \cdot \mathsf{len}(P_H) \cdot \kappa) = \Theta(\ell \log n + \kappa \log^2 n)$, where κ is security parameter. (The aforementioned complexity follows straightforwardly from [76].) Denote the function that the client and the server agreed to evaluation by $g(f_I, y)$, and let its BDD be P_g. One can easily integrate P_H and P_g to be P_p, where $g(f_{x_1}, \ldots, f_{x_m}, y) \leftarrow p(x_1, \ldots, x_m, y)$. The input

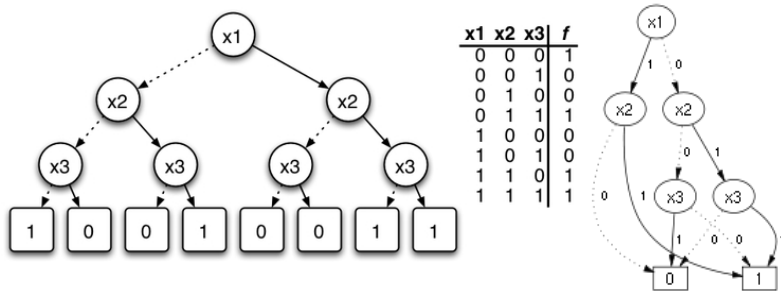


Figure 3.1: Disassembled CPIR and Boolean Function

of BDD P_g contains the bits of m documents $f_{x_1}, \ldots, f_{x_m}$ and y. The bits of the document f_{x_i} are computed at the server side, i.e., $h_1(x_i), \ldots, h_\ell(x_i)$. Instead of fetching and submitting the bit-wise encryptions of f_{x_i}, the client just submits the bit-wise encryptions of m indices. If a branch node in P_g depends on the j-th bit of document f_{x_i}, we can give the output of $h_j(x_i)$. In terms of PrivateBDD, we just replace the node of BDD P_g by the BDD of P_{h_j}.

Figure 3.2 shows an example of the BDD integration. The highlighted part of top-left BDD is a node $v_{f_{x,j}}$ with successors v_{V_1} and v_{T_2}, where the path is dependent on the value of $f_{x,j} \in \{0, 1\}$. The top-right P_{h_j} is the disassembled $(1, n)$-CPIR for the j-th bit of the documents. Instead of directly inputting $f_{x,j}$ to the top-left BDD, we insert P_{h_j} in between $v_{f_{x,j}}$ and its successors, i.e., replace the sink nodes of P_{h_j} from $\{0, 1\}$ to $\{T_2, V_1\}$ and route the initial node of P_{h_j} to $v_{f_{x,j}}$.

The new two-move generalized SPFE protocol is based on the integrated BDD. The client inputs $(x_1, \ldots, x_m)$ and y, and the server inputs a database $f = (f_0, \ldots, f_{n-1})$ and the function g. After the protocols execution, the client obtains $g(f_{x_1}, \ldots, f_{x_m}, y)$, and the server gets $\perp$.

Two-message BDD-based Generalized SPFE Protocol

Client's inputs: $x_1, \ldots, x_m, y$.
Server's inputs: $f = (f_0, \ldots, f_{n-1})$, function g.
Client's output: $g(f_{x_1}, \ldots, f_{x_m}, y)$.

1. The client and the server run in integrated PrivateBDD protocols P_p to compute the values $g(f_{x_1}, \ldots, f_{x_m}, y) \leftarrow p(x_1, \ldots, x_m, y)$.

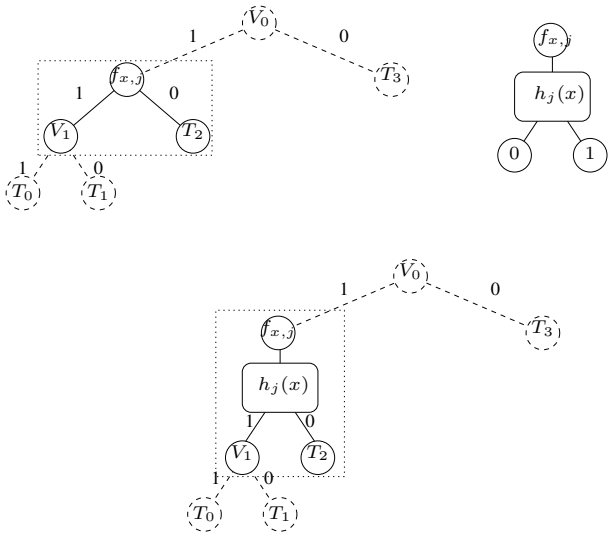


Figure 3.2: BDD integration

Efficiency. The server's online computation complexity of the new two-move generalized SPFE protocol is better than the two-move protocol in the paper [78] for any m. While the communication complexity is more or less the same. Table 3.1 shows the comparison of all 3 generalized SPFE protocols in this work.

Scheme	moves	Comm. Complexity	Comp. Complexity
S1 [78]	2	$\mathcal{O}((m \log n + \lambda) \cdot (\lambda' + \mathsf{len}(P_g) \cdot \kappa))$	$\mathcal{O}(\binom{n}{m} \cdot \mathsf{size}(P_g))$
S2 [78]	4	$\mathcal{O}(m(\log n + \ell + \kappa) + (m\ell + \lambda) \cdot (\lambda' + \mathsf{len}(P_g) \cdot \kappa))$	$\mathcal{O}(mn + \mathsf{size}(P_g))$
S3 3.2	2	$\mathcal{O}((m \log n + \lambda) \cdot (\lambda' + \log n \cdot \mathsf{len}(P_g) \cdot \kappa))$	$\mathcal{O}(\frac{n}{\log n} \cdot \mathsf{size}(P_g))$

Table 3.1: Comparison of 3 Generalized SPFE Protocols, where ℓ is the length of the documents f_i, λ is the length of y, and λ' is the output length of function g.

Theorem 2. *Let P_{h_j} be the BDD for function $f_{x_i,j} \leftarrow h_j(x_i)$, where $i \in [1, m]$ and $j \in [1, \ell]$. Let P_g be the BDD for function $g(f_{x_1}, \ldots, f_{x_m}, y)$ and P_p be the integrated BDD for function p such that $g(f_{x_1}, \ldots, f_{x_m}, y) \leftarrow p(x_1, \ldots, x_m, y)$. We have $\mathsf{len}(P_p) = \mathcal{O}(\mathsf{len}(P_H) \cdot \mathsf{len}(P_g))$ and $\mathsf{size}(P_p) = \mathcal{O}(\mathsf{size}(P_H) \cdot \mathsf{size}(P_g))$.*

Proof. During our integration, at most each layer of P_g will be inserted with P_{h_j} for corresponding j, therefore it is straightforward that $\mathsf{len}(P_p) = \mathcal{O}(\mathsf{len}(P_H) \cdot \mathsf{len}(P_g))$. In terms of size, we replace each unit evaluation of P_g by the evaluation of a PrivateBDD P_{h_j} for corresponding j, therefore it is also easy to see that $\mathsf{size}(P_p) = \mathcal{O}(\mathsf{size}(P_H) \cdot \mathsf{size}(P_g))$. □

The computation complexity of the aforementioned entire generalized SPFE protocol is $\mathcal{O}(\mathsf{size}(P_H) \cdot \mathsf{size}(P_g)) = \mathcal{O}(\frac{n}{\log n} \cdot \mathsf{size}(P_g))$ public-key operations. The communication complexity of the entire generalized SPFE protocol is *independent* to the document length ℓ. Assume $y \in \{0,1\}^{\lambda}$ and $p : \{0,1\}^{m*\lceil \log n \rceil + \lambda} \rightarrow \{0,1\}^{\lambda'}$. According to Theorem 1, it is easy to see that the communication complexity of our new generalized SPFE protocol is

$$\mathcal{O}((m \cdot \log n + \lambda) \cdot (\lambda' + \log n \cdot \mathsf{len}(P_g) \cdot \kappa)).$$

Security. Since the generalized SPFE protocol is a concrete application of two-move PrivateBDD protocol, the security follows the standard security of BDD-based cryptocomputing protocols, and we refer interested readers to [64, 76] for security discussion.

CHAPTER 4

OBLIVIOUS TRANSFER

In this chapter, we cover the work proposed by the author in paper "Simulatable Adaptive Oblivious Transfer With Statistical Receiver's Privacy" [99] and the unpublished work that was constructed by the author and his main supervisor in paper "A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument" [80].

4.1 Oblivious Transfer

4.1.1 Security Definition for Fully-Simulatable OT

To maintain consistency with earlier work, we recap security definition that used in papers [84, 23, 55]. Let tuple $(\mathcal{S}_I, \mathcal{R}_I, \mathcal{S}_T, \mathcal{R}_T)$ be the adaptive oblivious transfer $\mathsf{OT}^N_{k\times 1}$. Denote S_*, R_* as state values. During the initialization phase, the sender executes $S_0 \leftarrow \mathcal{S}_I(m_1, \ldots, m_N)$, and the receiver executes $R_0 \leftarrow \mathcal{R}_I()$. During the i-th transfer phase, $1 \leq i \leq k$, the sender executes $S_i \leftarrow \mathcal{S}_T(S_{i-1})$, and the receiver executes $(R_i, m^*_{\sigma_i}) \leftarrow \mathcal{R}_T(R_{i-1}, \sigma_i)$, where $1 \leq \sigma_i \leq N$ is the index of the message to be received. $m^*_{\sigma_i} = m_{\sigma_i}$ if successful, $m^*_{\sigma_i} = \perp$ if fails. The security of an $\mathsf{OT}^N_{k\times 1}$ scheme is defined in real-world/ideal-world paradigm with static corruption, i.e., the adversary $\mathcal{A}$ can only choose to corrupt either the sender or the receiver at the beginning of the experiments.

Real experiment. In experiment $\mathbf{Real}_{\hat{\mathbf{S}},\hat{\mathbf{R}}}(N, k, m_1, \ldots, m_N, \Sigma)$, the presumably cheating sender $\hat{\mathbf{S}}$ is given messages $(m_1, \ldots, m_N)$ as input and interacts with the presumably cheating receiver $\hat{\mathbf{R}}(\Sigma)$, where Σ is a selection algorithm that on input messages $m_{\sigma_1}, \ldots, m_{\sigma_{i-1}}$, outputs the index σ_i of the next message to be queried. In the initialization phase, $\hat{\mathbf{S}}$ and $\hat{\mathbf{R}}$ output the initial states S_0 and R_0. In the i-th transfer phase, for $1 \leq i \leq k$, the sender runs $S_i \leftarrow \hat{\mathbf{S}}(S_{i-1})$, and

the receiver runs $(R_i, m_i') \leftarrow \hat{\mathbf{R}}(R_{i-1})$, where m_i' is not necessary equal to m_i. After the k-th transfer, the output of the experiment $\mathbf{Real}_{\hat{\mathbf{S}},\hat{\mathbf{R}}}$ is the tuple (S_k, R_k).

We define the honest sender algorithm $\mathbf{S}$ as the one that runs $\mathcal{S}_I(m_1, \ldots, m_N)$ in the initialization phase, runs $\mathcal{S}_T()$ during each transfer phase, and returns $S_k = \varepsilon$ as its final output. The honest receiver algorithm $\mathbf{R}$ runs $\mathcal{R}_I()$ in the initialization phase, runs $\mathcal{R}_T(R_{i-1}, \sigma_i)$ during the i-th transfer phase, where the index σ_i is generated by Σ, and returns $R_k = (m_{\sigma_1}, \ldots, m_{\sigma_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathbf{S}}',\hat{\mathbf{R}}'}(N, k, m_1, \ldots, m_N, \Sigma)$, the presumably cheating sender $\hat{\mathbf{S}}'$ and the presumably cheating receiver $\hat{\mathbf{R}}'$ communicate with the ideal functionality $\mathcal{F}_{OT}^{N\times 1}$. In the initialization phase, $\hat{\mathbf{S}}'(m_1, \ldots, m_N)$ sends messages $m_1^*, \ldots, m_N^*$ to $\mathcal{F}_{OT}^{N\times 1}$. In the i-th transfer phase, for $1 \le i \le k$, $\hat{\mathbf{R}}'(\Sigma)$ sends to $\mathcal{F}_{OT}^{N\times 1}$ an index σ_i^*. $\mathcal{F}_{OT}^{N\times 1}$ then sends a tag 'Received' to $\hat{\mathbf{S}}'$, and $\hat{\mathbf{S}}'$ replies a bit $b_i \in \{0, 1\}$ to $\mathcal{F}_{OT}^{N\times 1}$. If $b_i = 1$ and $\sigma_i^* \in \{1, \ldots, N\}$, $\mathcal{F}_{OT}^{N\times 1}$ sends $m_{\sigma_i^*}^*$ to $\hat{\mathbf{R}}'$; otherwise, it sends $\perp$ to $\hat{\mathbf{R}}'$. After the k-th transfer, the output of the experiment $\mathbf{Ideal}_{\hat{\mathbf{S}}',\hat{\mathbf{R}}'}$ is the tuple (S_k, R_k).

Similarly, we define the honest sender algorithm $\mathbf{S}'(m_1, \ldots, m_N)$ as the one that sends $m_1, \ldots, m_N$ to $\mathcal{F}_{OT}^{N\times 1}$ in the initialization phase, sends $b_i = 1$ during each transfer phase, and returns $S_k = \varepsilon$ as its final output. The honest receiver $\mathbf{R}'$ submits the indices σ_i that generated by Σ to $\mathcal{F}_{OT}^{N\times 1}$, and returns $R_k = (m_{\sigma_1}, \ldots, m_{\sigma_k})$ as its final output.

Denote $\mathrm{poly}(\cdot)$ as a polynomially-bounded function. The security is defined as follows.

Sender Security. An $\mathsf{OT}_{k\times 1}^N$ is sender-secure if for every real-world p.p.t.-receiver $\hat{\mathbf{R}}$, there exists an ideal-world p.p.t.-receiver $\hat{\mathbf{R}}'$ such that for any $N = \mathrm{poly}(\kappa)$, any $k \in [0, N]$, any messages $(m_1, \ldots, m_N)$, any selection algorithm Σ, and every p.p.t.-distinguisher $\mathbf{D}$:

$$\mathbf{Real}_{\mathbf{S},\hat{\mathbf{R}}}(N, k, m_1, \ldots, m_N, \Sigma) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\mathbf{S}',\hat{\mathbf{R}}'}(N, k, m_1, \ldots, m_N, \Sigma)$$

Receiver Security. An $\mathsf{OT}_{k\times 1}^N$ is receiver-secure if for every real-world p.p.t.-sender $\hat{\mathbf{S}}$, there exists an ideal-world p.p.t.-sender $\hat{\mathbf{S}}'$ such that for any $N = \mathrm{poly}(\kappa)$, any $k \in [0, N]$, any messages $(m_1, \ldots, m_N)$, any selection algorithm Σ, and every p.p.t.-distinguisher $\mathbf{D}$:

$$\mathbf{Real}_{\hat{\mathbf{S}},\mathbf{R}}(N,k,m_1,\ldots,m_N,\Sigma) \quad \overset{c}{\approx} \quad \mathbf{Ideal}_{\hat{\mathbf{S}}',\mathbf{R}'}(N,k,m_1,\ldots,m_N,\Sigma)$$

Definition 20. We say a fully simulatable $\mathsf{OT}^N_{k\times 1}$ is secure iff it achieves both sender and receiver security.

4.1.2 Related Work and Our Results

Table 4.1 lists several existing $\mathsf{OT}^N_{k\times 1}$ schemes, together with our proposed schemes for comparison. In 2007, Camenisch, Neven and shelat [23] proposed an $\mathsf{OT}^N_{k\times 1}$ under the q-strong Diffie-Hellman and q-power Decisional Diffie-Hellman assumptions in bilinear groups. They used signatures as a key ingredient in their scheme. Later, Green and Hohenberger [53] showed an $\mathsf{OT}^N_{k\times 1}$ in the random oracle model under Decisional Bilinear Diffie-Hellman assumption. In their scheme, the sender encrypts message m_i by identity-based encryption under identity i. The receiver executes a blind key extraction protocol such that he/she can obliviously obtain the secret key of any identity. In 2008, Green and Hohenberger [54] introduced another OT that achieves UC security in the common reference string model, using the Groth-Sahai NIZK/NIWI proof [62] for pairing product equations. The scheme uses modified CL signature schemes [22], and its security is based on the decisional linear and q-Hidden LRSW assumptions. Jarecki and Liu [66] simplified the Camenisch *et al.* construction to a fully simulatable OT based on the Composite Decisional Residuosity and q-Decisional Diffie-Hellman Inversion assumptions. The scheme uses $pk = g^x$ and $c_i = m_i \cdot g^{1/(x+i)}$, where $g^{1/(x+i)}$ is Dodis-Yampolskiy verifiable pseudorandom function on input i [39]. The blind decryption is based on PRF with input i, and the protocol is the first efficient fully-simulatable OT without bilinear pairing. Rial, Kohlweiss and Preneel [91] presented an adaptive priced OT that achieves UC security using P-signature scheme [11]. In a priced OT, the receiver's privacy is still protected, even if the documents are paid at unique prices. The scheme is based on Decisional Linear, q-Triple Diffie-Hellman, and q-Hidden Strong Diffie-Hellman assumptions. Recently, Kurosawa and Nojima [69] gave adaptive OT constructions simple computational assumptions. Later, Kurosawa, Nojima and Phong [70] improved the scheme [69] by increasing the complexity of initialization phase. In 2011, Green and Hohenberger [55] proposed another fully simulatable OT under Decisional 3-party Diffie-Hellman assumption. Very recently, Kurosawa, Nojima and Phong [71] generalized their result in [70] to various schemes with different assumptions.

Scheme	Init. Cost	Tran. Cost	Assumption	Security
Folklore	—	$\mathcal{O}(N)$	general assumptions	Full Sim
[69]	$\mathcal{O}(N)$	$\mathcal{O}(N)$	Decisional n-th Residuosity/DDH	Full Sim
[84]	—	$\mathcal{O}(N)$	DDH + OT_1^2	Half Sim
[70]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH	Full Sim
[23]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	q-Power DDH + q-Strong DH	Full Sim
[54]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DLIN + q-Hidden LRSW	UC
[66]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	Comp. Dec. Residuosity + q-DDHI	Full Sim
[91]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DLIN + q-Hidden SDH + q-TDH	UC
[55]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	Decision 3-Party DH + DLIN	Full Sim
[71]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH/DLIN/DCR/QR/LWE	Full Sim
S1 [99]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	DDH	Full Sim
S2 [99]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	Decisional n-th Residuosity	Full Sim

Table 4.1: Survey on recent $\mathsf{OT}_{k\times 1}^N$ schemes. Ignore $\log N$ factor in communication complexity.

4.2 NIZK Argument for Shuffle

In our work [80], we construct a new computationally sound NIZK argument for correctness of shuffle. The only known NIZK argument for shuffle was proposed by Groth and Lu [60] in 2007. The Groth-Lu shuffle uses the BBS cryptosystem (where one ciphertext is 3 group elements), while we use the new lifted knowledge BBS cryptosystem (6 group elements). This difference however is quite small compared to our reduction of the argument size. On the other hand, our NIZK argument is proved to be computationally sound, instead of co-soundness that is used in Groth-Lu NIZK shuffle argument [60]. Our new shuffle argument is also based on more standard computational assumptions (PSDL and DLIN) compared to the Groth-Lu shuffle (PPA, SPA, and DLIN). Table 4.2 shows the comparison between our work [80] and Groth-Lu NIZK shuffle argument [60].

	Comm.	Comp.	CRS length	Assumption
[60]	$(15n + 120)$ G	$\Theta(n)$	$(2n + 8)$ G	PPA + SPA + DLIN
Scheme [80]	$(6n + 11)$ G	$\Theta(n)$	$(7n + 6)$ G	PKE + PSDL + DLIN

Table 4.2: Comparison of NIZK shuffle arguments

CHAPTER 5

PIR-WRITING PROTOCOL

In this chapter, we cover the work that was presented by the author and his supervisor in the paper "Two New Efficient PIR-Writing Protocols" [79], and we give an overview of the related topics.

5.1 PIR-writing, Oblivious Storage and Oblivious RAM

Remote outsourced database becomes increasingly popular, but the encryption of data itself cannot hide the access pattern to the outsourced database. Many closely related research topics are aiming to solve the problem, e.g., *Oblivious RAM* (ORAM), *Oblivious Storage* (OS) and *PIR-writing*. The notion of ORAM is introduced by Goldreich [48] in 1987, and it can be regarded as introducing "RAM model" to oblivious simulation of *Turing Machines* [89]. Later, Goldreich and Ostrovsky [49] investigated the problem of using the simulation of ORAM to hide access patterns in the context of RAM machines for software protection. Their motivation was to hide the program executed by a processor from an attacker snooping on the traffic to main memory.

RAM Model. The *Random Access Machine* (RAM) model consists of a CPU and a memory of size n. The CPU executes a program, accessing the memory via $\mathsf{Read}(i)$ and $\mathsf{Write}(i, v)$ operations, where $i \in [n]$ is an index to a memory location. $v \leftarrow \mathsf{Read}(i)$ fetches the value at memory location i, and $\mathsf{Write}(i, v)$ stores/updates new value v at memory location i. The access pattern of the program is defined as a sequence of read and write operations that the CPU makes to the memory, including both the data and the memory locations. In the implementation of a RAM, the CPU is a presumably probabilistic protected local device with limited number of registers. It accesses to an external device that simulates the memory and performs primitive arithmetic operations. We say that

a simulation is a secure ORAM if for any two access patterns in the ideal RAM, the corresponding access patterns in the simulation are computationally indistinguishable.

Recently, Boneh *et al.* introduced the notion of oblivious storage [20], which captures the security of remote storage. The client in oblivious storage corresponds to the CPU in ORAM, and the server in oblivious storage corresponds to the memory in ORAM. Ideally, the client storage should be constant and each "memory access" should be done in constant rounds. In ORAM, the cost of the traffic between the CPU and the memory is negligible, so the research task of ORAM simulation is to minimize the computational overhead. However, in oblivious storage, the round complexity of remote database access is an important factor in practice.

In terms of PIR-writing that is introduced by Boneh *et al.* [19], the round complexity is emphasized. Typically, the ideal PIR-writing protocol is just one-move (also known as half round), which consists only one message that the client sends to the server. The client outsources his encrypted database to the remote storage provider (the server). A PIR-writing protocol allows the client to update one element of the encrypted database such that the semi-honest server does not get to know which element was updated and to which value.

5.2 Security Definition of PIR-writing

The security definition of PIR-writing protocol in this work is formalized by the following PIR-writing game. Let $\mathcal{A}$ be a semi-honest probabilistic-polynomial time adversary (i.e.,, the server), and let $\mathcal{C}$ be the challenger. The game consists of the following steps:

1. The challenger $\mathcal{C}$ generates a pair of keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{G}(1^\kappa)$, and sends pk to $\mathcal{A}$.
2. $\mathcal{A}$ picks and sends to $\mathcal{C}$ a database $f = (f_0, \ldots, f_{n-1})$ of n elements with length ℓ.
3. $\mathcal{C}$ encrypts the database with pk and sends it back to $\mathcal{A}$.
4. (Challenge phase:) $\mathcal{A}$ picks and sends to $\mathcal{C}$ two index and value pairs $(x_0^*, y_0^*), (x_1^*, y_1^*)$. $\mathcal{C}$ picks $b_c \leftarrow \{0, 1\}$, and executes the PIR-writing protocol with input $(x_{b_c}^*, y_{b_c}^*)$, with $\mathcal{A}$ playing the role of the server.
5. $\mathcal{A}$ outputs her guess $b_c^* \in \{0, 1\}$ for b_c.

Note that here we do not have a query phase, since in our one-move protocols the malicious server can just execute the PIR-writing protocol with herself. Recall that the database she has is encrypted by using the client's public key.

Definition 21 (Client-privacy)**.** Let the adversary's advantage in the previous game be

$$\mathsf{Adv}_{\mathcal{A}}(1^{\kappa}) := \left| \Pr[b_c^* = b_c] - \frac{1}{2} \right| \ .$$

We say that a PIR-writing protocol is *client-private*, if for all p.p.t. adversaries $\mathcal{A}$, we have $\mathsf{Adv}_{\mathcal{A}}(1^{\kappa}) \leq \epsilon(\kappa)$, where $\epsilon(\cdot)$ is some negligible function.

5.3 Previous Works

Goldreich [48] gave two initial ORAM solutions: a "square-root solution" and a "recursive square-root solution". In the classical "square-root solution", the server memory is divided in two parts: the permuted memory and the shelter. The permuted memory stores n permuted client documents with a random permutation π and $\sqrt{n}$ dummy documents, while the shelter stores the records of recent accessed documents up to size $\sqrt{n}$. Each epoch consists of $\sqrt{n}$ memory access queries. To access memory location i, the client scans the entire shelter and checks whether the wanted document is in the shelter. If the document is not found in the shelter, the client retrieves document $\pi(i)$; otherwise the client retrieves document $\pi(n+i)$ (dummy document). The document is copied to the shelter, so the client updates the entire shelter to update the document. At the end of an epoch, the contents in the shelter is obliviously updated to the permuted memory, and re-shuffle the permuted memory with fresh permutation σ. Therefore, the sever storage is $n + 2\sqrt{n}$ in order to simulate ORAM with memory n. The computation overhead is $\mathcal{O}(\sqrt{n})$. The "recursive square-root solution" reduces the computation overhead to $\mathcal{O}(2^{\sqrt{\log n \log\log n}})$.

Ostrovsky [86] proposed a different ORAM scheme with "hierarchical solution" with computation overhead $\mathcal{O}(\min(\log^3 n, \log^3 t))$, where t is the program running time. The key technique is called oblivious shuffling for reshuffles. Later, a slightly different reshuffle method was introduced by Goldreich and Ostrovsky [49], but it behaves asymptotically the same as previous work. Williams and Sion [95] proposed a method to perform oblivious sorting to reduce the computation overhead to $\mathcal{O}(\log^2 n)$ with the client storage $\mathcal{O}(\sqrt{n})$. Later, Williams *et al.* [96] shows how to use Bloom Filters to reduce the computation overhead to $\mathcal{O}(\log n \log\log n)$ and the server storage to $\mathcal{O}(n)$. Pinkas and Reinman [88] revisited the problem and made use of cuckoo hashing to construct an ORAM scheme with constant client storage, $\mathcal{O}(n)$ server storage and computation overhead $\mathcal{O}(\log^2 n)$. In 2011, Goodrich and Mitzenmacher [52] showed an ORAM

scheme using cuckoo hashing with the client storage $\mathcal{O}(n^{\epsilon})$ and $\mathcal{O}(\log n)$ overhead. However, Kushilevitz *et al.* [73] showed that the hash-based ORAM scheme have security problem, and they proposed a security enhanced ORAM scheme with amortized overhead $\mathcal{O}(\log^2 n \log\log n)$ and $\mathcal{O}(1)$ client storage. Recently, Boneh *et al.* introduced the notion of oblivious storage [20] and proposed a new scheme with constant round complexity. Meanwhile, Ajtai [7] and Damgård *et al.* [38] investigated information-theoretical secure solution for ORAM with poly-logarithmic access overhead.

On the other hand, the first PIR-writing protocol was proposed Boneh *et al.* [19]. Essentially, it uses the BGN cryptosystem of [18] to send $2 \cdot \sqrt{n}$ ciphertexts c'_j and c''_j—such that the decryption of c_j is equal to the product of decryptions of c'_j and c''_j. Thus, this protocol has communication complexity $O(\sqrt{n})$ to modify one bit of a database. Clearly, by repeating the protocol, one will have a PIR-writing protocol with communication complexity $O(\ell \cdot \sqrt{n})$ for modifying ℓ bits. Later, Chandran *et al.* [26] proposed a way to decrease the communication complexity for larger ℓ. The solution consists of a pair of amortized protocols that have communication complexity $\mathcal{O}(\sqrt{\ell^{1+\alpha} \cdot n} \cdot \mathrm{polylog}(n))$ (where $\alpha \in (0,1)$ is a constant) for modifying ℓ bits of the database. The idea is to encode an n-bit database as a Dn-bit "virtual database" shared on M different servers by using unbalanced lossless expander graphs.

5.4 New PIR-writing Protocols

In Table 5.1, we compare two previously known sublinear-communication PIR-writing protocols, and our two new PIR-writing protocols that was proposed in [79]. Note that in the protocol from S1, the cost of a single public-key operation depends on ℓ, and on the number of the update. Since S2 is based on black-box usage of fully homomorphic encryption, it is easy to replace the fully homomorphic scheme with recent more efficient ones, e.g., [45].

Scheme	Communication Complexity	Computation Complexity	Security Assumption Size
[19]	$\mathcal{O}(\ell\sqrt{n})$	$\mathcal{O}(\ell n)$	subgroup decision [18] + polylog-communication CPIR [46, 75]
[26]	$\sqrt{\ell^{1+\alpha} \cdot n} \cdot \text{polylog}(n)$	$O(n \cdot \text{polylog}(n))$	subgroup decision [18] + polylog-communication CPIR [46, 75]
S1 [79]	$\Theta(u\ell \cdot \log n + u^2\kappa \cdot \log^2 n)$	$\mathcal{O}(n \cdot \log n)$	DCR assumption
S2 [79]	$\mathcal{O}(\kappa \log n + \kappa\ell)$	$\mathcal{O}(n \cdot \log n + \ell n)$	CPA-security of Gentry's fully-homomorphic cryptosystem, [44]

Table 5.1: Comparison of previous PIR-writing protocols and our two new protocols. In the case of the protocol from S1, u is the number of updates, and we give amortized communication over the first u updates. Note that the meaning of the unit computation depends on the protocol

CHAPTER 6

OBLIVIOUS DATABASE MANIPULATION

Most of the author's work is about two-party computation with computational security. (Note that there exists no nontrivial two party protocol with information-theoretical security.) In this chapter, we give overview of two papers [98, 74] that use multi-party computation for privacy preserving data-mining.

6.1 Frameworks for Share Computing

General Setup. A typical privacy-preserving data analysis application involves three types of entities: data donors, computing parties (referred to as *miners*), and clients. Data donors own sensitive data, miners gather the data and clients want to find various statistical results. In such a setting, privacy issues can be addressed with the client-server model formalized by Damgård and Ishai [35]. In this model, data donors submit their data in a secret shared form to the miner nodes, which later use share computing protocols to carry out the computations requested by the clients. Since data is secretly shared, individual records are protected as long as the miners do not form non-tolerated coalitions, i.e.,, they as a group follow the restrictions of share-computing protocols. Clients and data donors are not trusted and can arbitrarily violate protocol specifications. Depending on the underlying primitives and protocols, the framework can tolerate either semi-honest or malicious corruption of miners.

Share Computing Frameworks. A typical framework for multi-party computations is based on a secret sharing scheme and a set of protocols for manipulating the shares. A secret sharing scheme is specified by randomised sharing and recovery algorithms. The sharing algorithm splits a secret value $x \in \mathbb{Z}_N$ into shares

$x_1, \ldots, x_m$ that must be securely transferred to the miners $\mathcal{P}_1, \ldots, \mathcal{P}_m$, respectively. To recover the shared value, miners must together execute the reconstruction algorithm that takes in all shares and outputs the corresponding secret value. For example, the additive secret sharing scheme splits a secret x into shares such that the secret can be reconstructed by adding all the shares:

$$x \equiv x_1 + x_2 + \ldots + x_m \mod N \ .$$

The vector of shares $(x_1, x_2, \ldots, x_m)$ is commonly denoted by $[\![x]\!]$. Security properties of a secret sharing scheme are defined through a list of tolerable adversarial coalitions. Let $\mathcal{P}_{i_1}, \ldots, \mathcal{P}_{i_k}$ form a tolerable coalition, then the corresponding shares should leak nothing about the secret. More formally, the distributions $(x_{i_1}, \ldots, x_{i_k})$ and $(y_{i_1}, \ldots, y_{i_k})$ must coincide for any inputs $x, y \in \mathbb{Z}_N$.

Share computing protocols enable miners to perform oblivious computations with shares. For instance, miners can obtain a valid sharing of $x + y$ by locally adding their additive shares of x and y. Similarly, multiplying shares of x locally by a constant $\alpha \in \mathbb{Z}_N$ gives us shares of αx. A secret sharing scheme satisfying these two constraints is referred to as a *linear secret sharing scheme*. There are many linear secret sharing schemes, such as Shamir secret sharing scheme [92]. A share computing framework can be built on top of a linear secret sharing scheme by specifying a protocol for multiplication. Although protocols for secure share addition and multiplication are sufficient to achieve Turing completeness, the corresponding generic constructions are not efficient enough for practical applications. Hence, it makes sense to design specific protocols for common operations. In most cases, the effect of the network delay is several magnitudes larger than the time needed to deliver protocol messages. Thus, protocols with minimal round complexity are the most efficient. However, the round count is not an absolute measure, as the delivery time becomes dominant for large messages. Hence, overall communication complexity is also important. Table 6.1 depicts round complexity of various share computing operations implemented in Sharemind (c.f. Sect. 6.2.2, below), where ℓ is the bit length of modulus N. It is worth noting that logarithmic complexity in the bit size of the modulus N is asymptotically suboptimal, as theoretical construction by Damgård *et al.* [33] provides a constant round solution. However, the corresponding round count is larger than $\log \ell$ for all practical residue rings $\mathbb{Z}_N$.

For binary operations, we will use a shorthand $[\![x]\!] \circledast [\![y]\!]$ to denote the outputs of a share computing protocol that securely computes $x \circledast y$ from the shares of x and y.

Operation	Round count	Complexity
Multiplication	τ_{mul}	$\mathcal{O}(1)$
Coin-tossing	τ_{ct}	$\mathcal{O}(1)$
Smaller than	τ_{st}	$\mathcal{O}(\log \ell)$
Strictly less	τ_{sl}	$\mathcal{O}(\log \ell)$
Equality test	τ_{eq}	$\mathcal{O}(\log \ell)$
Bit-decomposition	τ_{bd}	$\mathcal{O}(\log \ell)$

Table 6.1: Round complexity of common share-computing operations

6.2 Practical MPC Systems

6.2.1 FairPlay

FairPlay is proposed by Malkhi *et al.* [81] at the Hebrew University of Jerusalem in 2004. FairPlay version 1.0 is designed for two-party computation against an active adversary based on Yao's garbled circuit [97]. A high-level program language called SFDL (Secure Function Definition Language) is designed to convert the complex function into a boolean circuit. The garbled circuit consists of permuted and encrypted truth tables and the two players will evaluate the optimized circuit on their private inputs. As standard procedure of Yao's garbled circuit, one player prepares and sends the circuit to the other player. The the other player runs OT for each input wire and evaluates the garbled circuit. In malicious setting, the player will verify $m - 1$ out of m garbled circuit a the beginning.

Late, Ben-David *et al.* [14] extended to FairPlayMP, which supports multi-party computation. It achieves an constant-round protocol (8 rounds) with the combination of the protocol by Beaver *et al.* [10] with the BGW [15] protocol. However, FairPlayMP is only secure against semi-honest adversaries in practice.

6.2.2 Sharemind

Sharemind is proposed by Bogdanov *et al.* [16] at University of Tartu and Cybernetica AS in 2008. Sharemind is currently used for *Privacy Preserving Data-Mining* (PPDM) with large scale databases. Current version of Sharemind runs between 3 parties (also called miners) that can tolerate at most one passively corrupted party. It works over 32-bit integer ring $\mathbb{Z}_{2^{32}}$. Therefore, the protocol design is slightly difficult due to the lack of multiplicative inverses. Subsequently, many existing protocols in the literature that are designed for fields do not work anymore, thus one has to design new protocols for Sharemind. It is implemented in C++, so it is very efficient (more than a million operations per second). Figure 6.1

shows the performance of elementary operations.

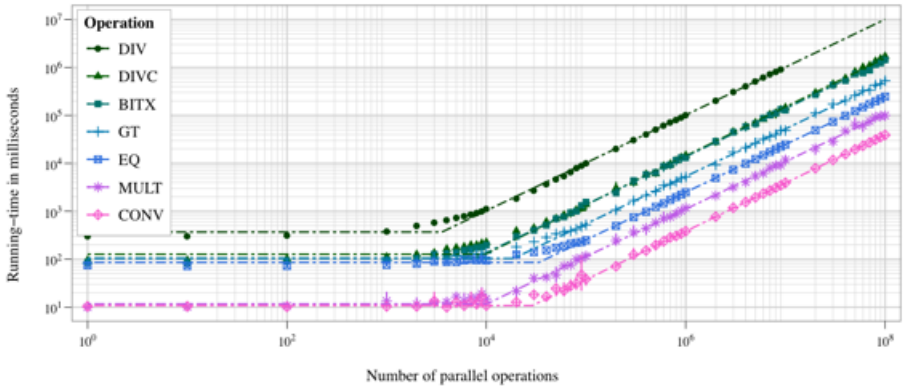


Figure 6.1: Sharemind Performance

Sharemind has its high-level language, SecreC (pronounced as "secrecy") and IDE. The syntax is very close to C without pointers. Many sophisticated PPDM algorithms have been developing on SecreC for Sharemind. Our work is also implemented and tested on Sharemind.

6.2.3 VIFF

VIFF (Virtual Ideal Functionality Framework) was proposed by Damgård *et al.* [34] at the University of Aarhus. VIFF is designed to work over asynchronous network. It automatically schedules the operations to be executed in parallel, minimizing average running time. VIFF is implemented in Python, using the Twisted network library. The data is shared by standard Shamir Secret Sharing Scheme [92] over some finite field $\mathbb{Z}_p$, and the number of parties is scalable. It provides many elementary operations, i.e., enough for the requirement of any function. Although, in theory, VIFF is secure against active adversaries, the practical implementation runs in passive setting for efficiency reason. There are a lot of applications developed over VIFF, such as distributed AES [37].

6.3 Oblivious Shuffle Result

In most cases, the effect of the network delay is several magnitudes larger than the time needed to deliver protocol messages. Thus, protocols with minimal round complexity are the most efficient. However, the round count is not an absolute

measure, as the delivery time becomes dominant for large messages. Hence, overall communication complexity is also important.

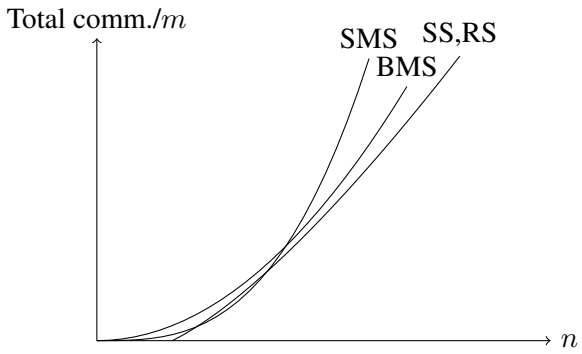
Figure 6.2 shows the expected performance of all three protocols from two aspects. Round complexity depicted on the bottom pane characterises the performance for small database sizes. For large databases, the running time is approximately linear to the communication complexity between individual miners. Hence, the comparison of communication complexities on the top pane is more relevant in practice.

Sort shuffle (, which is also known as Knuth shuffle [68],) is a shuffle that is implemented by sorting random numbers. Both simple and balanced matrix shuffles are implemented by permutation matrix. The difference between simple and balanced matrix shuffles is only the way to multiply permutation matrix. Resharing shuffle is our main contribution in paper [74], please refer [74] for the detail description.

Implementation. In practice, communication channels between miner nodes are commonly implemented using authenticated encryption and thus information-theoretical security is unachievable. Hence, the security level does not decrease if the group $\mathcal{C}$ agrees on a short random seed and later uses a secure pseudorandom generator to stretch locally into $\mathcal{O}(n \log n)$ bits needed for a random permutation. In particular, we can generate an array $f_{\mathsf{sk}}(1), \ldots, f_{\mathsf{sk}}(n)$ by applying a pseudorandom permutation f indexed by a seed sk. By sorting the array, we get a permutation that is computationally indistinguishable from a random permutation. We implemented the corresponding shuffle protocol for three miners by using 128-bit AES as f. For each mixing phase, $\mathcal{P}_a$ and $\mathcal{P}_b$ forming the group $\mathcal{C}$ exchanged 128-bit random sub-keys sk_a and sk_b and set $\mathsf{sk} \leftarrow \mathsf{sk}_a \oplus \mathsf{sk}_b$. The protocol was implemented in Sharemind framework using C++ programming language. The computing parties and the controller node ran on servers having two Intel Xeon X5670 2.93GHz processors and 48GB of RAM each. The servers were using Debian OS and were connected by gigabit Ethernet. Figure 6.3 shows the times required for oblivious shuffle of databases consisting of $10^3, \ldots, 10^7$ additively shared 32-bit integers.

6.4 Oblivious Sort Result

As regarding to general oblivious sorting algorithms, research interests are focus on how to achieve asymptotically optimal complexity, i.e., $\mathcal{O}(n \log n)$. AKS [6] sorting network is the first famous oblivious sorting network with computational complexity $\mathcal{O}(n \log n)$ compare-swap blocks. However, its constant factor is more than 6000 and it is quite complicated to implement. In practice, a lot of



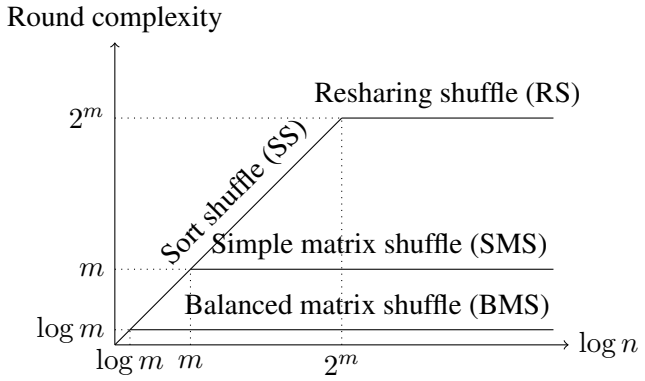


Figure 6.2: Performance comparison of oblivious shuffle protocols

well-known oblivious sorting algorithms with $\mathcal{O}(n \log^2 n)$ complexity are used in secure MPC systems, such as balanced sort [40] and Batcher's bitonic sort [9]. One can easily build an oblivious compare-swap block based on comparison circuit in MPC. So those algorithms require at least $\mathcal{O}(\log^2 n)$ rounds. For example, Kristján Valur Jónsson *et al.* [67] showed an oblivious sorting algorithm that is designed for MPC, and it uses $\mathcal{O}(n \log^2 n)$ comparisons in $\mathcal{O}(\log^2 n)$ rounds with practical constants. In 2009, a new simple oblivious sort called randomized Shellsort is introduced by Goodrich [51]. The sorting algorithm achieves $\mathcal{O}(n \log n)$ complexity, and it can finish with $\mathcal{O}(\log n)$ rounds. However, it is not guaranteed to sort; there is a small probability that the output array is not well sorted. In this work [98], we propose several constant-round oblivious sorting algorithms for

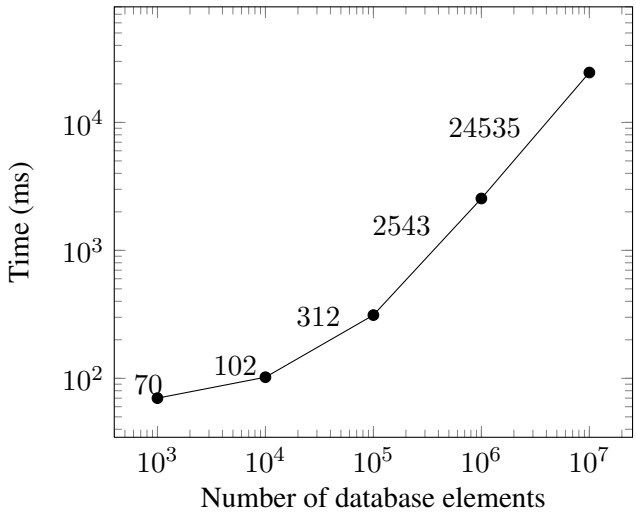


Figure 6.3: Performance of oblivious shuffle in three-party setting

MPC with different properties. In terms of oblivious shuffle, we give systematical solutions to address all possible cases. The maximum number of rounds is exponential to the number of parties, but independent of the input size (i.e., the number of documents). In practice, if the number of parties is 3 in semi-honest setting, the oblivious shuffle only takes 3 rounds. Table 6.2 illustrates the comparison between some widely used oblivious sorting algorithms and our schemes.

Sorting Scheme	Rounds	Comm. Complexity
AKS Sorting Network [6]	$O(\log n)$	$O(n \log n)$
Balanced Sort [40]	$O(\log^2 n)$	$O(n \log^2 n)$
Randomized Shellsort [51]	$O(\log n)$	$O(n \log n)$
Oblivious Sort for MPC [67]	$\mathcal{O}(\log^2 n)$	$\mathcal{O}(n \log^2 n)$
S 1 [98]	$O(1)$	$O(Rn)$
S 2 [98]	$O(1)$	$O(Rn)$
S 3 [98]	$O(1)$	$O(n^2)$

Table 6.2: Comparison of well-known oblivious sorting networks and our three new sorting algorithms for MPC. R is the range of numbers.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

In this work, we have proposed several cryptographic protocols for privacy and security of remote database. Those protocols are either more efficient then existing ones (under the same security definitions) or satisfy stronger security definitions. The focus of this work can be regarded as concrete topics in secure multi-party (two-party) computation. Our attention is mainly about improving the communication efficiency between the server(s) and the client(s). Additionally, non-interactive zero-knowledge proof and witness indistinguishable proof are also studied as primitive building blocks.

Some protocols are still not optimal, which leaves huge space for improvement. For example, the fully simulatable/UC-secure oblivious transfer protocol follows the framework that the server sends all the commitments (ciphertexts) of the entire database to the client in the initialization phase, so that the simulator can extract the server's input, i.e., the database. The limitation is that, according information theory, such OT must have $\Theta(n)$ communication, where n is the size of the database. We are going to investigate fully simulatable OT in malicious setting with poly-logarithmic communication complexity in our future research. PIR-writing protocols are also not optimal, so our next task is to construct the protocol with logarithmic (i.e., optimal) communication without using fully homomorphic encryption.

Bibliography

[1] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (1995), http://www.cdt.org/privacy/eudirective/EU_Directive_.html

[2] Abe, M.: Mix-Networks on Permutation Networks. In: Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security, ASIACRYPT '99. Lecture Notes in Computer Science (LNCS), vol. 1716, pp. 258–273. Springer-Verlag, London, UK (1999)

[3] Abe, M., Fehr, S.: Perfect NIZK with adaptive soundness. In: Proceedings of the 4th conference on Theory of cryptography, TCC'07. pp. 118–136. Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin, Heidelberg (2007)

[4] Abe, M., Hoshino, F.: Remarks on Mix-Network Based on Permutation Networks. In: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC '01. pp. 317–324. Lecture Notes in Computer Science (LNCS), Springer-Verlag, London, UK (2001)

[5] Adam, N.R., Worthmann, J.C.: Security-control methods for statistical databases: a comparative study. ACM Comput. Surv. 21, 515–556 (December 1989)

[6] Ajtai, M., Komlós, J., Szemerédi, E.: Sorting in c log n parallel steps. Combinatorica 3, 1–19 (January 1983)

[7] Ajtai, M.: Oblivious RAMs without cryptogrpahic assumptions. In: Proceedings of the 42nd ACM symposium on Theory of computing, STOC '10. pp. 181–190. Lecture Notes in Computer Science (LNCS), ACM, New York, NY, USA (2010)

[8] Bard, G.V., Courtois, N., Nakahara, J., Sepehrdad, P., Zhang, B.: Algebraic, AIDA/Cube and Side Channel Analysis of KATAN Family of Block Ci-

phers. In: Proceedings of the 11th International Conference on Cryptology in India, INDOCRYPT '10. LNCS, vol. 6498, pp. 176–196. Springer (2010)

[9] Batcher, K.E.: Sorting networks and their applications. In: Proceedings of the April 30–May 2, 1968, spring joint computer conference. pp. 307–314. AFIPS '68 (Spring), ACM, New York, NY, USA (1968)

[10] Beaver, D., Micali, S., Rogaway, P.: The Round Complexity of Secure Protocols. In: Proceedings of the 22nd annual ACM symposium on Theory of computing, STOC '90. pp. 503–513. ACM, New York, NY, USA (1990)

[11] Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Proceedings of the 5th conference on Theory of cryptography, TCC '08. pp. 356–374. LNCS, Springer-Verlag, Berlin, Heidelberg (2008)

[12] Bellare, M., Goldreich, O.: On Defining Proofs of Knowledge. In: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO' 92, LNCS, vol. 740, pp. 390–420. Springer Berlin / Heidelberg, London, UK (1993)

[13] Bellare, M., Palacio, A.: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In: Proceedings of the 24th Annual International Cryptology Conference, CRYPTO '04. LNCS, vol. 3152, pp. 273–289. Springer (2004)

[14] Ben-David, A., Nisan, N., Pinkas, B.: FairplayMP: a system for secure multi-party computation. In: Proceedings of the 15th ACM conference on Computer and communications security, CCS '08. pp. 257–266. LNCS, ACM, New York, NY, USA (2008)

[15] Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the 20th annual ACM symposium on Theory of computing, STOC '88. pp. 1–10. ACM, New York, NY, USA (1988)

[16] Bogdanov, D., Laur, S., Willemson, J.: Sharemind: A Framework for Fast Privacy-Preserving Computations. In: Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security, ESORICS '08. LNCS, vol. 5283, pp. 192–206. Springer-Verlag, Berlin, Heidelberg (2008)

[17] Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Proceedings of the 24th Annual International Cryptology Conference, Advances in

Cryptology, CRYPTO '04. LNCS, vol. 3152, pp. 227–242. Springer Berlin / Heidelberg (2004)

[18] Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Proceedings of Theory of Cryptography Conference 2005. LNCS, vol. 3378, pp. 325–342. Springer (2005)

[19] Boneh, D., Kushilevitz, E., Ostrovsky, R., III, W.E.S.: Public Key Encryption That Allows PIR Queries. In: Proceedings of the 27th annual international cryptology conference on Advances in cryptology, CRYPTO'07. pp. 50–67. LNCS, Springer-Verlag, Berlin, Heidelberg (2007)

[20] Boneh, D., Mazieres, D., Popa, R.A.: Remote Oblivious Storage: Making Oblivious RAM Practical. Computer Science and Artificial Intelligence Laboratory Technical Report (2011)

[21] Breitbart, Y., Hunt, III, H., Rosenkrantz, D.: On the size of binary decision diagrams representing Boolean functions. Theor. Comput. Sci. 145, 45–69 (July 1995)

[22] Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Proceedings of the 24th Annual International Cryptology Conference, Advances in Cryptology, CRYPTO '04. LNCS, vol. 3152, pp. 56–72. Springer (2004)

[23] Camenisch, J., Neven, G., Shelat, A.: Simulatable Adaptive Oblivious Transfer. In: Proceedings of the 26th annual international conference on Advances in Cryptology, EUROCRYPT '07. pp. 573–590. LNCS, Springer-Verlag, Berlin, Heidelberg (2007)

[24] Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067 (2000), http://eprint.iacr.org/

[25] Canetti, R., Ishai, Y., Kumar, R., Reiter, M.K., Rubinfeld, R., Wright, R.N.: Selective private function evaluation with applications to private statistics. In: Proceedings of the 20th annual ACM symposium on Principles of distributed computing, PODC '01. pp. 293–304. ACM, New York, NY, USA (2001)

[26] Chandran, N., Ostrovsky, R., Skeith, W.E.: Public-key encryption with efficient amortized updates. In: Proceedings of the 7th international conference on Security and cryptography for networks, SCN'10. pp. 17–35. LNCS, Springer-Verlag, Berlin, Heidelberg (2010)

[27] Chatfield, C., Hexel, R.: User identity and ubiquitous computing: User selected pseudonyms. In: Proceedings of Workshop on UbiComp Privacy (2005)

[28] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science, FOCS '95. pp. 41–57. IEEE Computer Society, Washington, DC, USA (1995)

[29] Cobham, A.: The recognition problem for the set of perfect squares. In: Proceedings of the 7th Annual Symposium on Switching and Automata Theory, SWAT '66. pp. 78–87. IEEE Computer Society, Washington, DC, USA (1966)

[30] Cramer, R., Damgård, I., MacKenzie, P.: Efficient Zero-Knowledge Proofs of Knowledge without Intractability Assumptions. In: Proceedings of the 3rd International Workshop on Practice and Theory in Public Key Cryptography, PKC '00, LNCS, vol. 1751, pp. 354–372. Springer-Verlag, London, UK (2000)

[31] Crépeau, C.: Equivalence Between Two Flavours of Oblivious Transfers. In: Proceedings of the Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology, CRYPTO '87. pp. 350–354. Springer-Verlag, London, UK (1987)

[32] Damgård, I.: Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '91. pp. 445–456. LNCS, Springer-Verlag, London, UK (1992)

[33] Damgård, I., Fitzi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation. In: Proceedings of the 3rd Theory of Cryptography Conference, TCC '06. LNCS, vol. 3876, pp. 285–304. Springer (2006)

[34] Damgård, I., Geisler, M., Krøigaard, M., Nielsen, J.B.: Asynchronous Multiparty Computation: Theory and Implementation. In: Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography, PKC '09. pp. 160–179. LNCS, Springer-Verlag, Berlin, Heidelberg (2009)

[35] Damgård, I., Ishai, Y.: Constant-Round Multiparty Computation Using a Black-Box Pseudorandom Generator. In: Proceedings of the 25th Annual

International Cryptology Conference for Advances in Cryptology, CRYPTO '05. LNCS, vol. 3621, pp. 378–394. Springer (2005)

[36] Damgård, I., Jurik, M.: A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System. In: Proceedings of 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC '01. pp. 119–136. LNCS, Springer-Verlag (2001)

[37] Damgård, I., Keller, M.: Secure Multiparty AES. In: Proceedings of the 14th International Conference for Financial Cryptography and Data Security, FC '10. LNCS, vol. 6052, pp. 367–374. Springer (2010)

[38] Damgård, I., Meldgaard, S., Nielsen, J.B.: Perfectly Secure Oblivious RAM without Random Oracles. In: Proceedings of the 8th conference on Theory of cryptography, TCC'11. pp. 144–163. LNCS, Springer-Verlag, Berlin, Heidelberg (2011)

[39] Dodis, Y., Yampolskiy, A.: A Verifiable Random Function with Short Proofs and Keys. In: Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, PKC '05, LNCS, vol. 3386, pp. 416–431. Springer Berlin / Heidelberg (2005)

[40] Dowd, M., Perl, Y., Saks, M., Rudolph, L.: The balanced sorting network. In: Proceedings of the 2nd annual ACM symposium on Principles of distributed computing, PODC '83. pp. 161–172. ACM, New York, NY, USA (1983)

[41] Even, S., Goldreich, O., Lempel, A.: A Randomized Protocol for Signing Contracts. Commun. ACM 28, 637–647 (June 1985)

[42] Furukawa, J.: Efficient and Verifiable Shuffling and Shuffle-Decryption. IEICE Transactions 88-A(1), 172–188 (2005)

[43] Furukawa, J., Sako, K.: An Efficient Scheme for Proving a Shuffle. In: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '01. pp. 368–387. LNCS, Springer-Verlag, London, UK (2001)

[44] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on Theory of computing, STOC '09. pp. 169–178. ACM, New York, NY, USA (2009)

[45] Gentry, C.: Fully Homomorphic Encryption without Bootstrapping. Cryptology ePrint Archive, Report 2011/277 (2011), `http://eprint.iacr.org/`

[46] Gentry, C., Ramzan, Z.: Single-Database Private Information Retrieval with Constant Communication Rate. In: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, ICALP '05. pp. 803–815. LNCS, Springer-Verlag (2005)

[47] Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Proceedings of the 43rd annual ACM symposium on Theory of computing, STOC '11. pp. 99–108. ACM, New York, NY, USA (2011)

[48] Goldreich, O.: Towards a theory of software protection and simulation by oblivious RAMs. In: Proceedings of the 19th annual ACM symposium on Theory of computing, STOC '87. pp. 182–194. ACM, New York, NY, USA (1987)

[49] Goldreich, O., Ostrovsky, R.: Software Protection and Simulation on Oblivious RAMs. Journal of the ACM 43, 431–473 (1996)

[50] Goldwasser, S.: Multi-Party Computations: Past and Present. In: Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing, PODC '97. pp. 1–6. ACM, New York, NY, USA (1997)

[51] Goodrich, M.T.: Randomized Shellsort: A Simple Oblivious Sorting Algorithm. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10. pp. 1262–1277. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2010)

[52] Goodrich, M.T., Mitzenmacher, M.: Privacy-Preserving Access of Outsourced Data via Oblivious RAM Simulation. In: Proceedings of the 38th International Colloquium for Automata, Languages and Programming, ICALP '11. LNCS, vol. 6756, pp. 576–587. Springer (2011)

[53] Green, M., Hohenberger, S.: Blind Identity-Based Encryption and Simulatable Oblivious Transfer. In: Proceedings of the 13th international conference on Theory and application of cryptology and information security: Advances in Crypotology, ASIACRYPT '07, LNCS, vol. 4833, pp. 265–282. Springer-Verlag, Berlin, Heidelberg (2007)

[54] Green, M., Hohenberger, S.: Universally Composable Adaptive Oblivious Transfer. In: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASIACRYPT '08. pp. 179–197. LNCS, Springer-Verlag, Berlin, Heidelberg (2008)

[55] Green, M., Hohenberger, S.: Practical Adaptive Oblivious Transfer from Simple Assumptions. In: Proceedings of the 8th conference on Theory of cryptography, TCC'11. pp. 347–363. Springer-Verlag, Berlin, Heidelberg (2011)

[56] Groth, J.: A Verifiable Secret Shuffle of Homomorphic Encryptions. In: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography, PKC '03. pp. 145–160. Springer-Verlag, London, UK, UK (2003)

[57] Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Proceedings of the 12th international conference on Theory and application of cryptology and information security: Advances in Crypotology, ASIACRYPT Õ06, LNCS, vol. 4284, pp. 444–459. Springer Berlin / Heidelberg (2006)

[58] Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: Proceedings of the 16th international conference on Theory and application of cryptology and information security: Advances in Crypotology, ASIACRYPT Õ10. LNCS, vol. 6477, pp. 321–340. Springer (2010)

[59] Groth, J., Ishai, Y.: Sub-linear Zero-Knowledge Argument for Correctness of a Shuffle. In: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology, EUROCRYPT '08. pp. 379–396. LNCS, Springer-Verlag, Berlin, Heidelberg (2008)

[60] Groth, J., Lu, S.: A Non-interactive Shuffle with Pairing Based Verifiability. In: Proceedings of the 13th international conference on Theory and application of cryptology and information security: Advances in Crypotology, ASIACRYPT '07. LNCS, vol. 4833, pp. 51–67. Springer Berlin / Heidelberg (2007)

[61] Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive Zaps and New Techniques for NIZK. In: Proceedings of the 26th Annual International Cryptology Conference: Advances in Cryptology, CRYPTO '06, LNCS, vol. 4117, pp. 97–111. Springer Berlin / Heidelberg (2006)

[62] Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology, EUROCRYPT '08, LNCS, vol. 4965, pp. 415–432. Springer-Verlag, Berlin, Heidelberg (2008)

[63] Hirt, M., Maurer, U.M.: Player Simulation and General Adversary Structures in Perfect Multiparty Computation. Journal of Cryptology 13(1), 31–60 (2000)

[64] Ishai, Y., Paskin, A.: Evaluating Branching Programs on Encrypted Data. In: Vadhan, S. (ed.) Proceedings of the 4th conference on Theory of cryptography, TCC '07, LNCS, vol. 4392, pp. 575–594. Springer-Verlag, Berlin, Heidelberg (2007)

[65] Jarecki, S.: Cryptographic primitives enforcing communication and storage complexity. In: Proceedings of the 6th international conference on Financial cryptography, FC '02. pp. 120–135. Springer-Verlag, Berlin, Heidelberg (2003)

[66] Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: Reingold, O. (ed.) Proceedings of the 6th conference on Theory of cryptography, TCC Õ09, LNCS, vol. 5444, pp. 577–594. Springer Berlin / Heidelberg (2009)

[67] Jónsson, K.V., Kreitz, G., Uddin, M.: Secure Multi-Party Sorting and Applications. Cryptology ePrint Archive, Report 2011/122 (2011), `http://eprint.iacr.org/`

[68] Knuth, D.: The Art of Computer Programming: Seminumerical algorithms, vol. 2. Addison-Wesley, Boston, 3 edn. (1998)

[69] Kurosawa, K., Nojima, R.: Simple Adaptive Oblivious Transfer without Random Oracle. In: Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASIACRYPT '09, LNCS, vol. 5912, pp. 334–346. Springer Berlin / Heidelberg (2009)

[70] Kurosawa, K., Nojima, R., Phong, L.T.: Efficiency-improved fully simulatable adaptive OT under the DDH assumption. In: Proceedings of the 7th international conference on Security and cryptography for networks, SCN '10. pp. 172–181. LNCS, Springer-Verlag, Berlin, Heidelberg (2010)

[71] Kurosawa, K., Nojima, R., Phong, L.T.: Generic Fully Simulatable Adaptive Oblivious Transfer. In: Proceedings of the 9th International Conference for Applied Cryptography and Network Security, ACNS '11. LNCS, vol. 6715, pp. 274–291. Springer (2011)

[72] Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science. pp. 364–373. IEEE Computer Society, Washington, DC, USA (1997)

[73] Kushilevitz, E., Lu, S., Ostrovsky, R.: On the (In)security of Hash-based Oblivious RAM and a New Balancing Scheme. Cryptology ePrint Archive, Report 2011/327 (2011), http://eprint.iacr.org/

[74] Laur, S., Willemson, J., Zhang, B.: Round-efficient Oblivious Database Manipulation. In: Proceedings of the 14th Information Security Conference, ISC '11. LNCS, Springer (2011)

[75] Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: Proceedings of the 8th International Conference for Information Security, ISC '05. Lecture Notes in Computer Science, vol. 3650, pp. 314–328. Springer (2005)

[76] Lipmaa, H.: First CPIR protocol with data-dependent computation. In: Proceedings of the 12th international conference on Information security and cryptology, ICISC'09. pp. 193–210. LNCS, Springer-Verlag, Berlin, Heidelberg (2010)

[77] Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. Cryptology ePrint Archive, Report 2011/009 (2011), http://eprint.iacr.org/

[78] Lipmaa, H., Zhang, B.: Efficient Generalized Selective Private Function Evaluation with Applications in Biometric Authentication. In: Proceedings of the 5th Information Security and Cryptology, Inscrypt '09. LNCS, vol. 6151, pp. 154–163. Springer (2009)

[79] Lipmaa, H., Zhang, B.: Two New Efficient PIR-Writing Protocols. In: Proceedings of the 8th International Conference on Applied cryptography and network Security, ACNS '10. LNCS, vol. 6123, pp. 438–455. Springer (2010)

[80] Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Cryptology ePrint Archive, Report 2011/394 (2011), http://eprint.iacr.org/2011/394

[81] Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay—a secure two-party computation system. In: Proceedings of the 13th conference on USENIX Security Symposium, SSYM '04. pp. 20–20. USENIX Association, Berkeley, CA, USA (2004)

[82] Nakahara, J., Sepehrdad, P., Zhang, B., Wang, M.: Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In: Proceedings of the 8th International Conference on Cryptology and Network Security, CANS '09. LNCS, vol. 5888, pp. 58–75. Springer-Verlag, Berlin, Heidelberg (2009)

[83] Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: Proceedings of the thirty-first annual ACM symposium on Theory of computing, STOC '99. pp. 245–254. ACM, New York, NY, USA (1999)

[84] Naor, M., Pinkas, B.: Oblivious Transfer with Adaptive Queries. In: Wiener, M. (ed.) Proceedings of the 19th Annual International Cryptology Conference: Advances in Cryptology, CRYPTO '99, Lecture Notes in Computer Science, vol. 1666, pp. 791–791. Springer Berlin / Heidelberg (1999)

[85] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM conference on Computer and Communications Security, CCS '01. pp. 116–125. ACM, New York, NY, USA (2001)

[86] Ostrovsky, R.: Efficient computation on oblivious RAMs. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. pp. 514–523. STOC '90, ACM, New York, NY, USA (1990)

[87] Pfitzmann, B., Schunter, M., Waidner, M.: Secure Reactive Systems. RZ 3206 (#93252), IBM Research Division, Zurich (May 2000), citeseer.ist.psu.edu/pfitzmann00secure.html

[88] Pinkas, B., Reinman, T.: Oblivious RAM Revisited. In: Proceedings of the 30th annual conference on Advances in cryptology, CRYPTO '10. pp. 502–519. Springer-Verlag, Berlin, Heidelberg (2010)

[89] Pippenger, N., Fischer, M.J.: Relations Among Complexity Measures. J. ACM 26, 361–381 (April 1979)

[90] Rabin, M.: How to exchange secrets by oblivious transfer. In: Technical Report TR-81. Aiken Computation Laboratory, Harvard University (1981)

[91] Rial, A., Kohlweiss, M., Preneel, B.: Universally Composable Adaptive Priced Oblivious Transfer. In: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography, Pairing '09. pp. 231–247. LNCS, Springer-Verlag, Berlin, Heidelberg (2009)

[92] Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979)

[93] Wegener, I.: Branching programs and binary decision diagrams: theory and applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)

[94] Wiesner, S.: Conjugate coding. Sigact News 15, 78–88 (1983)

[95] Williams, P., Sion, R.: Usable PIR. In: Proceedings of the Network and Distributed System Security Symposium, NDSS '08. The Internet Society (2008)

[96] Williams, P., Sion, R., Carbunar, B.: Building castles out of mud: practical access pattern privacy and correctness on untrusted storage. In: Proceedings of the 15th ACM conference on Computer and communications security, CCS '08. pp. 139–148. ACM, New York, NY, USA (2008)

[97] Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, FOCS '82. pp. 160–164. IEEE Computer Society, Washington, DC, USA (1982)

[98] Zhang, B.: Generic Constant-Round Oblivious Sorting Algorithm for MPC. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 240–256. Springer (2011)

[99] Zhang, B.: Simulatable Adaptive Oblivious Transfer With Statistical Receiver's Privacy. In: Proceedings of the 5th International Conference on Provable Security, ProvSec '11. LNCS, vol. 6980, pp. 52–67. Springer (2011)

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my main supervisor, Helger Lipmaa, whose help, stimulating suggestions and encouragement supported me during the time of research and the writing of this PhD thesis. I also want to thank my co-supervisor, Peeter Laud, for giving me many wise pieces of advice on the administrative issues during the past two years. I am also very thankful to my other coauthors Jan Willemson and Sven Laur for many interesting discussions that have taken place during the collaboration period.

Meanwhile, I want to thank all my other colleagues and friends and especially my parents, who have been a constant source of love, concern, support and strength throughout my studies. Please allow me to express my heartfelt gratitude to them. I would like to thank Madeline González Muñiz for proof-reading the introduction chapter of this thesis.

I was financially supported by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, by the Estonian Science Foundation grant no. 8058 and no. 8124, by EU FP6-IST-15964: “AEOLUS", by the Estonian ICT Doctoral School, by the Tiger University Program of the Estonian Information Technology Foundation and, last but not least, Cybernetica AS, one of the few private research companies in Estonia, where I have also been employed as a researcher for two years.

KOKKUVÕTE
(SUMMARY IN ESTONIAN)

EFEKTIIVSED KRÜPTOGRAAFILISED PROTOKOLLID TURVALISTE JA PRIVAATSETE KAUGANDMEBAASIDE JAOKS

Pilvearvutus on üks valdkondi, mida turuliidrid, nagu näiteks Apple, Google või Microsoft, tänapäeval märkimisväärselt edendavad ja reklaamivad. Kogu IT-tööstus on liikumas traditsioonilisest mudelist, kus asutused ja organisatsioonid haldavad oma andmebaase ise, mudelisse, kus andmebaaside hoidmine ning opereerimine on allhankega üle antud kolmandate osapoolte vastutusalasse. Kolmandatele osapooltele üleantud andmebaasid on eriti populaarsed mobiilsete seadmete, nagu nutitelefonid ja võrguarvutid, juures, mis ei suuda suuri andmehulki salvestada. Mitmed uued mõisted, nagu näiteks "tarkvara kui teenus", võimaldavad ettevõtetel pakkuda kasulikku funktsionaalsust kaugsalvestatud andmetel. Selline "pilvine ilm" toob aga endaga kaasa ka palju uusi turva- ja privaatsusprobleeme kaugandmebaasidel.

Käesolevas töös uuritakse põhiliselt krüptograafilistel protokollidel põhinevaid lahendusi nendele kaugandmebaaside turva- ja privaatsusprobleemidele. Klassikaliselt on krüptograafilisi protokolle kasutatud konfidentsiaalsus-, terviklus- ja autentsusomaduste saavutamiseks. Tänapäevased protokollid suudavad aga enamat, olles võimelised arvutitevahelise koostöö jaoks tagama palju erinevaid kasulikke omadusi. Käesolevas töös konstrueerime krüptograafilisi protokolle kindlate ülesannete jaoks järgnevas kolmes stsenaariumis.

Esimeses stsenaariumis haldab kaugserver oma enda privaatset, tundlike kirjetega andmebaasi ning kliendid esitavad sellele andmebaasile päringuid. Me uuri-

me juhtu, kus klient küsib mitut andmebaasikirjet. Server lubab tal nende kirjete väärtuseid teada saada, kuid ainult nende ning mitte teiste kirjete omi. Samal ajal soovib klient, et server ei saaks teada, milliseid kirjeid ta päris. Lahendame selle probleemi adaptiivse, täissimuleeritava komponeeritava peitedastuse (*Oblivious Transfer*) abiga. Me konstrueerime ka uue mitteinteraktiivse nullteadmustõestuse kaardipaki segamise korrektsusele; sel tulemusel on ka sõltumatu väärtus teistes olukordades. Samuti uurime juhtu, kus klienti ei huvita mitte üksikud kirjed serveri hallatavast andmebaasist, vaid teatud statistilised andmed, mis seda andmebaasi iseloomustavad. Selle juhu jaoks konstrueerime me üldistatud selektiivse privaatse funktsiooniväärtustusprotokolli, mis säilitab nii kliendi kui ka serveri privaatsuse.

Teises stsenaariumis kuulub privaatne andmebaas kliendile, kes on selle haldamiseks ja päringuvastusteks üle andnud serverile (kaugsalvestusteenuse pakkujale). Selles stsenaariumis kliendid reeglina servereid ei usalda ja seega krüpteerivad nad andmebaasi kirjed enne nende serverile üleandmist. Selles stsenaariumis läheb tarvis protokolli, mis lubab kliendil mõnda andmebaasielementi uuendada, nii et server ei saa aru millist kirjet uuendatakse ning ei saa teada ka uut kirje väärtust. Käesolevas töös pakume me välja kaks uut kommunikatsiooniefektiivset protokolli sellise privaatse uuendamise jaoks eeldusel, et server viib läbi ainult *passiivseid* ründeid, s.t. ta järgib protokolli, kuid kliendilt saadud teadetest üritab välja lugeda andmeid, mida ta teada saama ei peaks.

Kolmandas stsenaariumis kogutakse tundlikke andmeid paljudelt erinevatelt isikutelt ja organisatsioonidelt ning salvestatakse need andmebaasis, mida hallatakse mitme serveri koostöös viisil, et ründaja kontrolli all olev(ad) server(id) ei suuda teatud turvaeelduste kehtimisel andmebaasi sisu kohta mitte midagi teada saada. Sellises stsenaariumis kasutatakse andmebaasi jagamiseks mitme serveri vahel tavaliselt mingit ühissalastusskeemi; andmebaasipäringute tegemiseks ning statistiliste andmete leidmiseks tuleb kasutada turvalise ühisarvutuse tehnikaid. Me pakume selle stsenaariumi jaoks välja mitmeid erinevaid kasulikke ning väheste kommunikatsiooniraundidega protokolle. Need protokollid on kasutatavad näiteks kirjete peitvalimiseks, filtreerimiseks, sorteerimiseks ja segamiseks — privaatsust säilitava andmekaevanduse põhioperatsioonideks.

Esimene peatükk annab ülevaate töö probleemipüstitusest, sellega seonduvat tausta ning meie panust selle lahendamisse. Teine peatükk tutvustab täpsemalt edasise töö mõistmiseks vajalikke eelteadmiseid ja kasutatavat terminoloogiat. Kolmandas peatükis defineerime formaalselt selektiivse privaatse funktsiooniväärtustusprotokolli ja parandame artiklis "Efektiivne üldistatud selektiivse privaatsuse funktsiooniväärtustusprotokoll rakendustega biomeetrilises tuvastuses" meie poolt varem esitatud protokolli. Neljandas peatükis tutvustame N elemendist k peiteedastuse probleemi ning esitame mitteinteraktiivse nullteadmustõestuse kaardipaki segamise protokollile, millel on ka sõltumatu väärtus. Peatüki si-

su põhineb peamiselt artiklitel “Simuleeritav Adaptiivne peiteedastus statistilise vastuvõtjaprivaatsusega” ning “Veelgi efektiivsem arvutuslikult terviklik mitte-interaktiivne nullteadmusel põhinev segamistõestus”. Viies peatükk seletab privaatset andmete uuendamist mitteusaldatud serveris ning annab ülevaate artiklis “Kaks uut efektiivset PIR-kirjutamise protokolli” meie poolt välja pakutud lahendustest, andes ülevaate ka lähedalt seotud peidetud muutmälu probleemist. Kuues peatükk tutvustab mitme osapoole ühisarvutuse mudelit ning privaatsust säilitava andmekaeve ülesannet. Seejärel anatkse ülevaade selle lahendamisel tehtud edusammudest, mida kajastati ka artiklites “Üldine konstantse raundide arvuga peidetud sorteerimise algoritm mitme osapoole ühisarvutuste jaoks” ning “Raundide arvu mõttes efektiivne andmebaasi peidetud muutmine”. Seitsmes peatükk kätkeb endas kokkuvõtet, kus tutvustakse ka mõningaid huvitavaid lahtiseks jäävaid küsimusi.

ORIGINAL PUBLICATIONS

Publication	Pages
Helger Lipmaa, Bingsheng Zhang, "Efficient Generalized Selective Private Function Evaluation with Applications in Biometric Authentication"	77 – 86
Helger Lipmaa, Bingsheng Zhang, "Two New Efficient PIR-Writing Protocols"	89 – 106
Bingsheng Zhang, "Generic Constant-Round Oblivious Sorting Algorithm for MPC"	109 – 125
Sven Laur, Jan Willemson, Bingsheng Zhang, "Round-efficient Oblivious Database Manipulation"	129 – 155
Bingsheng Zhang, "Simulatable Adaptive Oblivious Transfer With Statistical Receiver's Privacy"	159 – 174
Helger Lipmaa, Bingsheng Zhang, "A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument'	177 – 198

CURRICULUM VITAE

Personal data

Name	Bingsheng Zhang
Birth	December 14, 1984 Hangzhou, China
Citizenship	Chinese
Languages	Chinese, English
Address	Narva mnt. 165A-6 Tartu Estonia
Contact	cellphone: +372 58156022 email: zhang@ut.ee

Education

2009–	University of Tartu, Estonia, Ph.D. candidate
2007–2008	University College London, UK, M.Sc. in Information security
2003–2007	Zhejiang University of Technology, China, B.Eng. in Computer science and Technology
2003–2007	Zhejiang University of Technology, China, B.L. in Law (second degree)
2000–2003	Hangzhou No.4 High School, China, secondary education

Employment & Internship

2011–	University of Tartu, Estonia, Teaching Assistant
2009– 2011	Cybernetica AS, Estonia, researcher

2008–2009	British Telecommunications plc, UK, internship
2008	University College London, UK, research associate (part-time)
2007	Zhejiang Province Advanced People's Court, China, internship

ELULOOKIRJELDUS

Isikuandmed

Nimi	Bingsheng Zhang
Sünniaeg ja -koht	14. detsember 1984 Hangzhou, Hiina
Kodakondsus	Hiina
Keelteoskus	hiina, inglise
Aadress	Narva mnt. 165A-6 Tartu Estonia
Kontaktandmed	tel. +372 58156022 e-post zhang@ut.ee

Haridustee

2009–	Tartu Ülikool, doktorant
2007–2008	University College London, infoturbemagister
2003–2007	Zhejiangi Tehnoloogiaülikool, arvutiteaduse ja -tehnoloogia bakalaureus
2003–2007	Zhejiangi Tehnoloogiaülikool, õigusteaduse bakalaureus
2000–2003	Hangzhou 4. keskkool, keskharidus

Teenistuskäik

2011–	Tartu Ülikool, informaatika assistent
2009–2011	Cybernetica AS, teadur
2008–2009	British Telecommunications plc, praktikant

2008	University College London, teaduslik töötaja
2007	Zhejiangi provintsi kõrgem rahvakohus, praktikant

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

1. **Mati Heinloo.** The design of nonhomogeneous spherical vessels, cylindrical tubes and circular discs. Tartu, 1991, 23 p.
2. **Boris Komrakov.** Primitive actions and the Sophus Lie problem. Tartu, 1991, 14 p.
3. **Jaak Heinloo.** Phenomenological (continuum) theory of turbulence. Tartu, 1992, 47 p.
4. **Ants Tauts.** Infinite formulae in intuitionistic logic of higher order. Tartu, 1992, 15 p.
5. **Tarmo Soomere.** Kinetic theory of Rossby waves. Tartu, 1992, 32 p.
6. **Jüri Majak.** Optimization of plastic axisymmetric plates and shells in the case of Von Mises yield condition. Tartu, 1992, 32 p.
7. **Ants Aasma.** Matrix transformations of summability and absolute summability fields of matrix methods. Tartu, 1993, 32 p.
8. **Helle Hein.** Optimization of plastic axisymmetric plates and shells with piece-wise constant thickness. Tartu, 1993, 28 p.
9. **Toomas Kiho.** Study of optimality of iterated Lavrentiev method and its generalizations. Tartu, 1994, 23 p.
10. **Arne Kokk.** Joint spectral theory and extension of non-trivial multiplicative linear functionals. Tartu, 1995, 165 p.
11. **Toomas Lepikult.** Automated calculation of dynamically loaded rigid-plastic structures. Tartu, 1995, 93 p, (in Russian).
12. **Sander Hannus.** Parametrical optimization of the plastic cylindrical shells by taking into account geometrical and physical nonlinearities. Tartu, 1995, 74 p, (in Russian).
13. **Sergei Tupailo.** Hilbert's epsilon-symbol in predicative subsystems of analysis. Tartu, 1996, 134 p.
14. **Enno Saks.** Analysis and optimization of elastic-plastic shafts in torsion. Tartu, 1996, 96 p.
15. **Valdis Laan.** Pullbacks and flatness properties of acts. Tartu, 1999, 90 p.
16. **Märt Põldvere.** Subspaces of Banach spaces having Phelps' uniqueness property. Tartu, 1999, 74 p.
17. **Jelena Ausekle.** Compactness of operators in Lorentz and Orlicz sequence spaces. Tartu, 1999, 72 p.
18. **Krista Fischer.** Structural mean models for analyzing the effect of compliance in clinical trials. Tartu, 1999, 124 p.

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
20. **Jüri Lember.** Consistency of empirical k-centres. Tartu, 1999, 148 p.
21. **Ella Puman.** Optimization of plastic conical shells. Tartu, 2000, 102 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
25. **Maria Zeltser.** Investigation of double sequence spaces by soft and hard analitical methods. Tartu, 2001, 154 p.
26. **Ernst Tungel.** Optimization of plastic spherical shells. Tartu, 2001, 90 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 p.
28. **Rainis Haller.** *M(r,s)*-inequalities. Tartu, 2002, 78 p.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
30. **Eno Tõnisson.** Solving of expession manipulation exercises in computer algebra systems. Tartu, 2002, 92 p.
31. **Mart Abel.** Structure of Gelfand-Mazur algebras. Tartu, 2003. 94 p.
32. **Vladimir Kuchmei.** Affine completeness of some ockham algebras. Tartu, 2003. 100 p.
33. **Olga Dunajeva.** Asymptotic matrix methods in statistical inference problems. Tartu 2003. 78 p.
34. **Mare Tarang.** Stability of the spline collocation method for volterra integro-differential equations. Tartu 2004. 90 p.
35. **Tatjana Nahtman.** Permutation invariance and reparameterizations in linear models. Tartu 2004. 91 p.
36. **Märt Möls.** Linear mixed models with equivalent predictors. Tartu 2004. 70 p.
37. **Kristiina Hakk.** Approximation methods for weakly singular integral equations with discontinuous coefficients. Tartu 2004, 137 p.
38. **Meelis Käärik.** Fitting sets to probability distributions. Tartu 2005, 90 p.
39. **Inga Parts.** Piecewise polynomial collocation methods for solving weakly singular integro-differential equations. Tartu 2005, 140 p.
40. **Natalia Saealle.** Convergence and summability with speed of functional series. Tartu 2005, 91 p.
41. **Tanel Kaart.** The reliability of linear mixed models in genetic studies. Tartu 2006, 124 p.
42. **Kadre Torn.** Shear and bending response of inelastic structures to dynamic load. Tartu 2006, 142 p.

43. **Kristel Mikkor.** Uniform factorisation for compact subsets of Banach spaces of operators. Tartu 2006, 72 p.
44. **Darja Saveljeva.** Quadratic and cubic spline collocation for Volterra integral equations. Tartu 2006, 117 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
46. **Annely Mürk.** Optimization of inelastic plates with cracks. Tartu 2006. 137 p.
47. **Annemai Raidjõe.** Sequence spaces defined by modulus functions and superposition operators. Tartu 2006, 97 p.
48. **Olga Panova.** Real Gelfand-Mazur algebras. Tartu 2006, 82 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
50. **Margus Pihlak.** Approximation of multivariate distribution functions. Tartu 2007, 82 p.
51. **Ene Käärik.** Handling dropouts in repeated measurements using copulas. Tartu 2007, 99 p.
52. **Artur Sepp.** Affine models in mathematical finance: an analytical approach. Tartu 2007, 147 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
54. **Kaja Sõstra.** Restriction estimator for domains. Tartu 2007, 104 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
57. **Evely Leetma.** Solution of smoothing problems with obstacles. Tartu 2009, 81 p.
58. **Ants Kaasik.** Estimating ruin probabilities in the Cramér-Lundberg model with heavy-tailed claims. Tartu 2009, 139 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
60. **Indrek Zolk.** The commuting bounded approximation property of Banach spaces. Tartu 2010, 107 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
63. **Marek Kolk.** Piecewise Polynomial Collocation for Volterra Integral Equations with Singularities. Tartu 2010, 134 p.

64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
65. **Larissa Roots.** Free vibrations of stepped cylindrical shells containing cracks. Tartu 2010, 94 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
68. **Olga Liivapuu.** Graded q-differential algebras and algebraic models in noncommutative geometry. Tartu 2011, 112 p.
69. **Aleksei Lissitsin.** Convex approximation properties of Banach spaces. Tartu 2011, 107 p.
70. **Lauri Tart.** Morita equivalence of partially ordered semigroups. Tartu 2011, 101 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.
74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
75. **Nadežda Bazunova.** Differential calculus $d^3 = 0$ on binary and ternary associative algebras. Tartu 2011, 99 p.
76. **Natalja Lepik.** Estimation of domains under restrictions built upon generalized regression and synthetic estimators. Tartu 2011, 133 p.