NANYANG
TECHNOLOGICAL
UNIVERSITY

A STUDY OF PRIVATE INFORMATION RETRIEVAL AND RELATED PRIMITIVES

LIANG FENG ZHANG

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2011

A Study of Private Information Retrieval and Related Primitives

Liang Feng Zhang

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2011

Acknowledgments

This thesis was done under the supervision of Associate Professor Yeow Meng Chee and Associate Professor Huaxiong Wang. I am greatly indebted to them for providing me with the opportunity to conduct the research. Their deep intuition and highly insightful comments regarding the problems I worked on influenced me a lot. The research topic proposed by Prof. Wang allows me to enter the great field of computer science. His continuous encouragement in every stage of my research was indeed a great help. I benefit a lot from the extremal set theory and the probabilistic method that Prof. Chee taught. I benefit even more from his style of teaching which is extremely clear and illuminating.

I am indebted to Amos Beimel, Tao Feng and San Ling for their collaborations that led to several of the results discussed in this thesis.

Publications

- Yeow Meng Chee, Tao Feng, San Ling, Huaxiong Wang, and Liang Feng Zhang. Query-efficient locally decodable codes of subexponential length. In Computational Complexity. To Appear, 2011.

- Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. Oblivious transfer and n-variate linear function evaluation. In Bin Fu, Ding-Zhu Du (eds.) COCOON 2011. LNCS, vol. 6842, pp. 627-637. Springer, Heidelberg, 2011.

- Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. On Bringer-Chabanne EPIR protocol for polynomial evaluation. In Journal of Mathematical Cryptology, To Appear, 2011.

- Amos Beimel, Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. Communication-efficient distributed oblivious transfer. In Journal of Computer and System Sciences. To Appear, 2012.

Contents

Abstract

In this thesis, we study four notions related to Private Information Retrieval (PIR), namely Extended Private Information Retrieval (EPIR), Locally Decodable Codes (LDCs), Distributed Oblivious Transfer (DOT) and Oblivious Linear Function Evaluation (OLFE).

EPIR allows a user to evaluate a function on one of the data blocks of a server, in such a way that the server learns no information on neither the function nor the identity of the accessed data block, and the user cannot learn more information on the data blocks except the evaluation. Bringer and Chabanne (2009) proposed an EPIR protocol where the user's function is a polynomial over a finite field and the server's data blocks are elements of the field. In Chapter 3, we show that their protocol fails frequently when one coefficient of the user's polynomial is primitive in the finite field and the others belong to the prime subfield of the field. Our results call for new EPIR protocols that avoid the above deficiency.

LDCs allow one to encode a message as a codeword such that each symbol of the message can be recovered by a probabilistic decoding algorithm which only looks at a small number of coordinates of the codeword, even if a constant fraction of the codeword has been corrupted. Efremenko (2009) proposed a framework for constructing constant-query LDCs of subexponential length. Within this framework, LDCs of query complexity as small as 3 can be obtained. These codes depend on composite numbers which have nice algebraic properties. In Chapter 4, we present our discovery of a new class of algebraically nice numbers. In particular, we show that *Mersenne numbers* (numbers of the form $M_t = 2^t - 1$, where t is a prime) which are products of two primes are algebraically nice. These numbers enable us to improve the known

constructions of LDCs and PIR protocols.

DOTs are oblivious transfers in the distributed setting where the receiver can learn a secret of the sender with the help of intermediate servers. Compared with the classical oblivious transfers, DOTs are computationally efficient and achieve information-theoretic privacy. However, the known DOT protocols have linear communication complexity between the receiver and servers, which may be prohibitive when the sender has a huge number of secrets. In Chapter 5, we propose both a specific reduction from DOT to polynomial interpolation-based PIR and a general reduction from DOT to any PIR. Our reductions yield the first DOT protocols of sublinear communication complexity between the receiver and servers.

OLFE is a cryptographic primitive between two parties where one party has a multivariate linear function and the other party wants to evaluate the function on a private input. In Chapter 6, we introduce this primitive and present its interesting applications in the cryptographic study and protocol design. We give an unconditionally secure and fully simulatable reduction from classical OT to OLFE. Specifically, we show that any classical OT can be unconditionally securely reduced to a number of invocations of a given OLFE except for a negligible failure probability. Using this reduction, we show that any classical OT can be efficiently reversed in the sense that the roles of the sender and the receiver are interchanged.

Chapter 1

Introduction

In this thesis, we study four notions related to private information retrieval, namely extended private information retrieval, locally decodable codes, distributed oblivious transfer and oblivious linear function evaluation.

1.1 Extended Private Information Retrieval

Private information retrieval (PIR) was introduced by Chor et al. [26] in 1995. As a well-known cryptographic primitive, it has attracted a large amount of attention in the cryptographic community. Various models for PIR have been developed during the past 16 years.

Information-theoretic private information retrieval. A *t-private k-server* information-theoretic private information retrieval protocol [26] allows a user $\mathcal{U}$ to retrieve a data item (or a bit) x_i from k servers $\mathcal{S}_1, \ldots, \mathcal{S}_k$, each of which has a copy of a database $x \in \{0,1\}^n$, such that any coalition of up to t servers learns absolutely no information on which item (i.e., the value of i) $\mathcal{U}$ is interested in. What makes the retrieval problem nontrivial is the privacy requirement we impose on i. This requirement is often termed as *user privacy*. Without the user privacy, $\mathcal{U}$ could send i to a single server in plain form and receive x_i immediately. This *no-privacy* solution only requires $\mathcal{U}$ to exchange $\log n + 1$ bits with the server. On the other hand, if there is only one server and the user privacy should be met, then $\Omega(n)$ bits [27] are

necessarily exchanged. We see a tradeoff between the user privacy and communication cost. Actually, the most challenging problem involving PIR is to decide the optimal communication cost while preserving the user privacy. The communication cost, often termed as *communication complexity*, is formally defined to be the total number of bits exchanged between the user and servers for retrieving one bit. Due to the $\Omega(n)$ lower bound [27] for the single server case (i.e. $k = 1$), the only hope for a solution of communication complexity $o(n)$ is to have multiple servers (i.e. $k > 1$). Since 1995, numerous constructions [26, 1, 7, 5, 92, 96, 36, 55, 22] of multi-server PIR protocols have been proposed. The most efficient 1-private 2-server and 3-server PIR protocols are of communication complexity $O(n^{1/3})$ [26] and $\exp(O(\sqrt{\log n \log\log n}))$ [36], respectively. For $t > 1$, the t-private k-server PIR protocols constructed by [5, 92] are well-known and of communication complexity $O\left(n^{1/\lfloor(2k-1)/t\rfloor}\right)$. Barkol et al. [2] suggested one compose a t_1-private k_1-server PIR protocol and a t_2-private k_2-server PIR protocol in order to obtain a t_1t_2-private k_1k_2-server PIR protocol. Specifically, by composing the most efficient 1-private 3-server PIR protocol [36] with itself t times, one can obtain a t-private 3^t-server PIR protocol of communication complexity $\exp(O(\sqrt{\log n \log\log n}))$ for every integer $t \geq 2$.

Computationally private information retrieval. Kushilevitz et al. [62] showed that the linear communication complexity of single-server PIR protocols can be avoided if the server is computationally bounded (i.e., running in polynomial time) and certain cryptographic assumptions hold. They constructed a single-server PIR protocol of communication complexity $O(n^\epsilon)$ under the assumption that deciding quadratic residuosity problem is hard, where $\epsilon > 0$ is an arbitrarily small constant. The single-server PIR protocol [62] is also known as a *computationally private information retrieval* (CPIR) protocol because the user privacy only holds when the server is computationally bounded. Following [62], a number of much more efficient CPIR protocols have been proposed. Specifically, Cachin et al. [18] constructed a CPIR protocol of communication complexity $O(\log^8 n)$ under the Φ-hiding assumption. So far, the most efficient CPIR protocol was obtained by Gentry et al. [40] under the assumption that the decision subgroup problem is hard. It requires the user to ex-

change $O(k+d)$ bits with the server for retrieving d bits, where $k \geq \log n$ is the security parameter. Other constructions of CPIR can be found in [63, 93, 21, 50]. CPIR is an important primitive in the theory of cryptography. It has been shown to imply one way functions [6], collision-resistant hash functions [53] and oblivious transfers [31]. These results show that CPIR is among the "hard" primitives which belong to the public-key world. On the other hand, it is not "harder" than trapdoor permutations (TDP) due to the construction [63] of CPIR out of TDP.

Symmetrically private information retrieval. The notion of PIR does not impose any privacy requirement on the database $x \in \{0,1\}^n$. An honest-but-curious user may obtain many bits after executing a PIR protocol. For example, the well-known 1-private 2-server PIR protocol [26] based on covering code allows a user to learn as many as $O(n^{1/3})$ bits in each execution. In order to avoid this deficiency, Gertner et al. [42] introduced the notion of *data privacy* and proposed transformations from information-theoretic PIR protocols to the so-called *symmetrically private information retrieval* (SPIR) protocols. The SPIR protocols allow a user to learn only one bit per execution. They are said to meet the data privacy requirement. SPIR can be defined in the computational setting as well. Following the definition of secure computation [44], in the computational setting, a PIR protocol is said to achieve data privacy if, for a query, the user cannot tell whether it is interacting with a server having the database or a simulator only knowing the retrieved bit. Naor et al. [68] proposed transformations from PIR to SPIR in the computational setting.

Extended private information retrieval. *Extended private information retrieval* (EPIR) was motivated by privacy-preserving biometric authentication and formally defined by Bringer et al. [16]. It enables a user to privately evaluate a fixed and public function with two inputs, one chosen block from a database and one additional string. Recently, EPIR was generalized by Bringer et al. [15] in order to attain more flexibility. In the generalized setting, the function to be evaluated is neither fixed nor public. Instead, it is chosen from a set of public functions by the user. A new EPIR protocol for polynomial evaluation in the generalized setting was proposed in [15]. EPIR is indeed a combination of private information retrieval [26] and secure

two-party computation [44, 94]. More precisely, it is a generalization of SPIR in the computational setting. EPIR is also connected to selective private function evaluation [20], oblivious polynomial evaluation [71] and private keyword search [38].

1.2 Locally Decodable Codes

A classical error-correcting code allows one to encode a message as a codeword such that the message can be recovered even if the codeword gets corrupted in a number of coordinates. However, to recover even a small fraction of the message, one has to consider all or most of the coordinates of the received (possibly corrupted) codeword. Katz et al. [57] considered error-correcting codes where each symbol of the message can be probabilistically recovered by looking at a limited number of coordinates of a corrupted codeword. Such codes are known as *locally decodable codes* (LDCs). While LDCs were formally defined by Katz et al. [57], the explicit discussion of the notion of them dates back to [64, 86, 79], even as early as the work of Reed and Muller [82] in 1950s. Informally, a (k, δ, ϵ)-LDC $\mathbf{C} : \Sigma^n \to \Gamma^N$ (Σ and Γ are any finite alphabets) encodes a message x as a codeword $\mathbf{C}(x)$ such that each symbol x_i can be recovered with probability at least $1 - \epsilon$, by a probabilistic decoding algorithm that makes at most k queries, even if the codeword is corrupted in up to δN locations. LDCs have many applications in cryptography and complexity theory [39, 87], and have attracted a considerable amount of attention [33, 74, 59, 35, 89, 45, 85, 81, 92, 58, 96, 36, 48, 55, 34, 61, 97].

For constant δ and ϵ, the efficiency of a (k, δ, ϵ)-LDC $\mathbf{C} : \Sigma^n \to \Gamma^N$ is measured by its *length* N and *query complexity* k. Ideally, we want both N and k to be as small as possible. Katz et al. [57] proved that there do not exist families of 1-query LDCs. Goldreich et al. [45] obtained an exponential lower bound of $\exp(\Omega(n))$ on the length of 2-query *linear* LDCs. Kerenidis et al. [59] showed that the optimal length of *any* 2-query LDCs is $\exp(O(n))$ via a quantum argument. For k-query ($k \geq 3$) LDCs, Woodruff [91] obtained a superlinear lower bound of $\Omega(n^{(k+1)/(k-1)}/\log n)$ on their length. Other lower bounds have been obtained by [33, 74, 35, 89, 85].

It has been conjectured for a long time that the length of any constant-query LDC should exponentially depend on its message length. This conjecture was disproved by Yekhanin [96], who constructed a 3-query LDC of length $\exp(\exp(O(\log n/\log\log n)))$ under the assumption that there are infinitely many *Mersenne primes* (primes of the form $M_t = 2^t - 1$, where t is prime). Subsequently, Yekhanin's construction was nicely reformulated by Raghavendra [81] using group homomorphism. Efremenko [36] generalized Yekhanin's construction and established a framework for constructing LDCs in which the assumption on Mersenne primes is no longer necessary. Efremenko [36] constructed a k_r-query ($k_r \leq 2^r$) LDC of length $N_r = \exp(\exp(O(\sqrt[r]{\log n(\log\log n)^{r-1}})))$ for every integer $r \geq 2$, and in particular, a 3-query ($k_2 = 3$) LDC of length N_2. The main ingredient of Efremenko's construction is Grolmusz's super-polynomial size set-systems with restricted intersections [49]. These set-systems depend on certain composite numbers, which have significant impact on the query complexity of the resulting LDCs. Efremenko [36] showed that the set-systems based on 511 can result in a 3-query LDC of length N_2, where the query complexity 3 is strictly less than 2^2 ($r = 2$). We say that the number 511 is *algebraically nice*. It turns out that 511 is the only algebraically nice number before our work in this thesis. In fact, Efremenko [36] left it open to find more algebraically nice numbers.

Itoh et al. [55] developed a composition method in Efremenko's framework, which allows one to compose Efremenko's k_r-query ($k_r \leq 2^r$) LDC of length N_r and k_l-query ($k_l \leq 2^l$) LDC of length N_l to obtain a k-query LDC of length N_{r+l} such that $k \leq k_r k_l$. For every integer $r \geq 4$, taking Efremenko's 3-query LDC and k_{r-2}-query LDC as building blocks, the composition method yields a k-query LDC of length N_r in which $k \leq 3 \cdot 2^{r-2}$, improving the query complexity of Efremenko's LDC of the same length by a factor of 3/4. The improvement is due to the 3-query LDC. Hence, it is of great interest to have as many such 3-query LDCs as possible, or equivalently, as many algebraically nice numbers as possible.

1.3 Distributed Oblivious Transfer

Oblivious transfer. *Oblivious transfer* (OT) [80, 37, 13] is an important cryptographic primitive and has numerous applications in protocol design [95, 46, 60]. Rabin's OT [80] allows a sender Alice to send a bit b to a receiver Bob such that with probability 0.5 Bob obtains the bit, and with the same probability he does not, while Alice does not know which event has occurred. The $\binom{n}{1}$-OT [13, 37] involves a sender Alice who has n secrets $s_1, \ldots, s_n$ and a receiver Bob who has a choice $c \in [n]$. It allows Bob to obtain s_c and no more information while c is not revealed to Alice. Many variants of OT [17, 14, 9, 3, 12] have been defined in the literature. For example, in an xor oblivious transfer, the sender Alice has two bits b_0, b_1 and the receiver Bob is allowed to obtain b_c for a choice $c \in \{0, 1, \oplus\}$, where $b_c = b_0 \oplus b_1$ if $c = \oplus$. An execution of the xor oblivious transfer gives Alice nothing about c and Bob no more information than b_c.

As many other primitives, OT cannot be constructed from scratch [76, 29]. They are necessarily built on various cryptographic assumptions and cannot achieve unconditional security for both the sender and the receiver. On the one hand, OT has been built on specific assumptions such as the hardness of general integer factoring, Diffie-Hellman assumptions, quadratic residuosity assumption, and worst-case lattice assumptions [80, 8, 70, 56, 78, 77]. On the other hand, the results of [37, 51] show that OT can be built on general assumptions such as the existence of (dense) trapdoor permutations. Due to the well-known results [43, 52], it looks hard to build OT on assumptions as weak as the existence of one way functions.

Distributed oblivious transfer. OT built on specific assumptions are computationally secure and involves heavy public-key operations. Naor et al. [69] introduced *distributed oblivious transfer* (DOT) which is information-theoretically secure and computationally efficient. Subsequently, their DOT was generalized by [11, 73]. Let t, τ, k, l be positive integers such that $0 \leq t, \tau \leq k \leq l$. Informally, a (t, τ)-private (k, l)-DOT-$\binom{n}{1}$ involves a sender $\mathcal{D}$ who has n secrets $W_1, \ldots, W_n$, a receiver $\mathcal{U}$ who wants to learn W_i, and l servers $\mathcal{S}_1, \ldots, \mathcal{S}_l$. It consists of a *setup stage* when the

sender distributes the secrets among the l servers, and a *transfer stage* when the receiver contacts k out of the l servers. Such a protocol should satisfy: (i) after the transfer stage, any coalition of up to t servers learns no information on i; (ii) before the transfer stage, the receiver learns no information on the secrets, even if it colludes with up to τ servers; (iii) after the transfer stage, a malicious receiver is able to obtain at most one of the secrets, even if it colludes with up to τ malicious servers.

Privacy preserving architecture. DOT has applications in the *privacy preserving architecture* of Naor et al. [72]. The architecture involves an *auction issuer*, several *auctioneers* and numerous *bidders*. In privacy preserving auctions, the auctioneers publish the details of the auctions they are organizing, receive both encrypted bids from the bidders and garbled programs from the auction issuer, and then compute and publish the auctions. The privacy of the bidders is preserved as long as the auction issuer and auctioneers do not collude. DOT provides solutions for privacy preserving auction. In particular, the bidders may play the role of the receiver in DOT. Therefore, DOT protocols of small communication overhead are preferred.

In the remaining of this section, we review several notions related to DOT.

Random server model. In PIR and SPIR, the database is replicated among non-communicating servers. This *data replication problem* may be serious because each server could be broken into. To get around this problem, Gertner et al. [41] proposed PIR in the *random server model* that consists of a *setup stage* when the database owner shares the database among random severs and an *on-line stage* when the user retrieves a data item by interacting with the random servers.

Commodity-based model. A PIR in the *commodity-based model* [30] involves a user who wants to retrieve x_i of a database $x \in \{0,1\}^n$, a number of database servers which have x and *commodity servers* which send off-line messages (called *commodities*). It consists of an *off-line commodity stage* when the commodity servers send commodities to the user and database servers and an *on-line retrieval stage* when the user retrieves x_i by interacting with the database servers. The off-line commodities help to reduce the on-line communication involving the user sharply.

Oblivious polynomial evaluation. *Oblivious polynomial evaluation* [68] is a

primitive between a sender Alice who has a polynomial $f(x)$ and a receiver Bob who wants to evaluate $f(\alpha)$ for a private input α. It requires that Alice cannot learn α and Bob learns no more information on $f(x)$ except $f(\alpha)$. The oblivious polynomial evaluation can be computationally reduced to OT [68].

Private information storage. *Private information storage* [75] allows users to read and write into a database which is shared among non-communicating servers such that each individual server learns absolutely no information on which location the users are reading or writing, and what the users are writing.

Trusted initializer model. A $\binom{n}{1}$-OT in the *trusted initializer model* [83] involves a sender who has n secrets, a receiver who wants to learn one of the secrets and a trusted initializer. It consists of a *setup stage* when the trusted initializer sends private information to the sender and receiver, and a *request-reply stage* when the receiver learns one secret by interacting with the sender.

1.4 Oblivious Linear Function Evaluation

We study a new cryptographic primitive called *oblivious n-variate linear function evaluation with choice space $\mathcal{C}$* ($\mathcal{C}$-OLFE$_n$) [24] in Chapter 6. The primitive is always associated with a finite field $\mathbb{F}$ and involves a sender Alice who has an n-variate linear function $s(x) = \sum_{i=1}^{n} s_i x_i \in \mathbb{F}[x]$ and a receiver Bob who has a choice $c \in \mathcal{C} \subseteq \mathbb{F}^n$. It allows Bob to evaluate $s(c)$ in such a way that Alice cannot learn c and Bob learns no more information on $s(x)$ except $s(c)$. The $\mathcal{C}$-OLFE$_n$ captures a variety of well-known cryptographic primitives. For any integer $n \geq 2$, the $\binom{n}{1}$-OT with $s = (s_1, \ldots, s_n) \in \mathbb{F}^n$ as Alice's input is a $\mathcal{C}$-OLFE$_n$ where $\mathcal{C} = \{(1, 0, \ldots, 0), \ldots, (0, 0, \ldots, 0, 1)\}$ and $s(x) = \sum_{i=1}^{n} s_i x_i$. The oblivious polynomial evaluation with $f(x) = \sum_{i=0}^{n-1} f_i x^i \in \mathbb{F}[x]$ as Alice's input is a $\mathcal{C}$-OLFE$_n$ where $s(x) = f(x)$ and $\mathcal{C} = \{(1, \alpha, \ldots, \alpha^{n-1}) : \alpha \in \mathbb{F}\}$. The xor oblivious transfer with (b_0, b_1) as Alice's input is a $\mathcal{C}$-OLFE$_n$ where $s(x) = b_0 x_0 \oplus b_1 x_1 \in \mathbb{F}_2[x]$ and $\mathcal{C} = \{(1, 0), (0, 1), (1, 1)\}$.

The $\mathcal{C}$-OLFE$_n$ turns out to be an interesting stepstone in protocol design. In particular, it has an interesting application in the problem of reversing OT. Crépeau

[28] raised the question of whether it is possible to implement OT in one direction using several invocations of OT in the other. He proved that a $\binom{2}{1}$-OT from Alice (as a sender) to Bob (as a receiver) can be reduced to $4k$ invocations of a $\binom{2}{1}$-OT from Bob (as a sender) to Alice (as a receiver) except for a failure probability 2^{-k}. Wolf et al. [90] proposed a method of reversing the $\binom{2}{1}$-OT which requires only 1 invocation of the given $\binom{2}{1}$-OT.

1.5 Results of the Thesis

We summarize the main results of this thesis in this section. The related notations can be found in respective chapters.

- In Chapter 3, we show that the Bringer-Chabanne EPIR protocol for polynomial evaluation fails frequently for an infinite class of polynomials:

  Bringer-Chabanne EPIR is Not Correct [23]. A user with input $(F(t), i) \in L[t] \times [N]$ does not learn the expected result (i.e., $F(R_i)$) with non-negligible probability if one of the coefficients of $F(t)$ is primitive in L and the others belong to the prime subfield of L.

- In Chapter 4, we study the algebraically nice numbers which yield 3-query LDCs in Efremenko's framework and show that:

  Algebraically Nice Mersenne Numbers [22]. If m is a Mersenne number and the product of two primes, then it is algebraically nice.

- In Chapter 5, we propose both a specific reduction and a general reduction from DOT to information-theoretic PIR:

  Specific Reduction [4]. For integers $t, \tau, k \leq l$ such that $1 \leq t \leq k-1$ and $0 \leq \tau \leq k-1-t$, there is a (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ of communication complexity $O\left(n^{1/\lfloor (k-1-\tau)/t \rfloor}\right)$ between a semi-honest receiver and servers.

  General Reduction [4]. For integers $t, \tau, k \leq l$ such that $1 \leq t \leq k-2$ and $0 \leq \tau \leq k-1-t$, if there is a t-private $(k-t-\tau)$-server PIR protocol of

communication complexity $\gamma(n)$, then there is a (t, τ)-private (k, l)-DOT-$\binom{n}{1}$ of communication complexity $O(\gamma(n))$ between the receiver and servers.

- In Chapter 6, we present an unconditionally secure reduction from $\binom{n}{1}$-OT to the newly defined primitive $\mathcal{C}$-OLFE$_n$:

  Reducing OT to OLFE [24]. Any $\binom{n}{1}$-OT can be unconditionally securely reduced to kn invocations of a given $\mathcal{C}$-OLFE$_n$ except for a negligible failure probability 2^{-k}, where $\mathcal{C}$ contains all unit vectors of length n.

Chapter 2

Preliminaries

2.1 Mathematical Notions

2.1.1 Standard Mathematical Notations

We use the following standard mathematical notations:

- $[m] = \{1, 2, \ldots, m\}$;
- $\mathbb{Z}_m$ is the ring of integers modulo m;
- $\mathbb{Z}_m^*$ is the set of units in $\mathbb{Z}_m$;
- $\langle x, y \rangle_m \equiv \sum_{i=1}^{h} x_i y_i \bmod m$ is the dot product of $x, y \in \mathbb{Z}_m^h$;
- $\mathbb{F}_q$ is the finite field of order q;
- $\mathbb{F}_q^*$ is the multiplicative group of $\mathbb{F}_q$;
- $d_H(x, y)$ is the Hamming distance between vectors x and y;
- $\mathbb{N}$ is the set of all nonnegative integers;
- $\mathbb{Z}$ is the ring of integers;
- $\mathbb{C}$ is the field of complex numbers;
- $\mathbb{C}^*$ is the multiplicative group of $\mathbb{C}$.

2.1.2 Group Rings, Characters and Cyclotomic Cosets

We adopt most notions from [32, 88, 66, 67] in this section. Let G be a finite abelian group with multiplication being its group operation. The *group ring*

$$\mathbb{Z}[G] = \left\{ \sum_{g \in G} a_g g : a_g \in \mathbb{Z} \right\}$$

is a ring of formal sums, in which addition and multiplication are defined as follows:

$$A + B = \sum_{g \in G} (a_g + b_g) g, \qquad A \cdot B = \sum_{g \in G} \sum_{h \in G} a_g b_h gh,$$

where $A = \sum_{g \in G} a_g g, B = \sum_{g \in G} b_g g \in \mathbb{Z}[G]$. We also use the following standard notations in $\mathbb{Z}[G]$:

$$A^{(j)} = \sum_{g \in G} a_g g^j, \ \forall j \in \mathbb{Z}, \qquad D = \sum_{g \in D} g, \ \forall D \subseteq G.$$

A group homomorphism $\chi : G \to \mathbb{C}^*$ is called a *character* of G. If $|G| = n$, then G has n characters. Let $\widehat{G}$ be the multiplicative group of all characters of G where $\chi_1\chi_2(g) = \chi_1(g)\chi_2(g)$ for any $\chi_1, \chi_2 \in \widehat{G}$ and $g \in G$. The identity χ_0 of $\widehat{G}$ maps every $g \in G$ to 1. For every $\chi \in \widehat{G}$, the *order* of χ is the least positive integer l such that $\chi^l = \chi_0$. Every $\chi \in \widehat{G}$ can be easily extended to $\mathbb{Z}[G]$ in a way such that $\chi(A) = \sum_{g \in G} a_g \chi(g)$ for every $A = \sum_{g \in G} a_g g \in \mathbb{Z}[G]$. It is well-known that $\sum_{g \in G} \chi(g) = 0$ whenever $\chi \in \widehat{G}$ is not equal to χ_0 and $\chi(A^{(-1)}) = \overline{\chi(A)}$ for every $\chi \in \widehat{G}$ and $A \in \mathbb{Z}[G]$, where $\overline{\chi(A)}$ is the complex conjugate of $\chi(A)$.

Let q be a prime power and m be a positive integer such that $\gcd(q, m) = 1$. For every $s \in \mathbb{Z}_m$, the *cyclotomic coset of q modulo m containing s* is defined to be

$$\mathbb{E}_s = \{(sq^l \bmod m) \in \mathbb{Z}_m : l = 0, 1, \ldots\},$$

where s is the smallest number in $\mathbb{E}_s$ and called *coset representative* of $\mathbb{E}_s$. It is clear that all distinct cyclotomic cosets of q modulo m form a partition of $\mathbb{Z}_m$.

2.1.3 Distribution Ensembles

We denote by $w \leftarrow \mathbb{W}$ the experiment of randomly choosing an element from a set $\mathbb{W}$. Let W and W' be random variables with domain $\mathbb{W}$. The *statistical distance* between W and W' is defined to be

$$\mathcal{SD}(W, W') = \frac{1}{2} \sum_{w \in \mathbb{W}} \Big| \Pr[W = w] - \Pr[W' = w] \Big|.$$

The random variables W and W' are said to be identical and denoted by $W \equiv W'$ if $\mathcal{SD}(W, W') = 0$. Let $W^\star$ be a random variable with domain $\mathbb{W}^\star$. For any $w \in \mathbb{W}$ and $w^\star \in \mathbb{W}^\star$ such that $\Pr[W^\star = w^\star] > 0$, we denote by $\Pr[W = w | W^\star = w^\star]$ the conditional probability that $W = w$ given $W^\star = w^\star$. We shall write $\Pr[W = w | w^\star]$ instead of $\Pr[W = w | W^\star = w^\star]$ whenever $W^\star$ is clear from the context. A *distribution ensemble* $X = \{X(k, a)\}_{k \in \mathbb{N}, a \in D}$ is an infinite sequence of probability distributions, where each distribution $X(k, a)$ is associated with an integer $k \in \mathbb{N}$ and a finite string $a \in D$ for some domain D. (Typically, $D = \{0, 1\}^*$.) A function $\delta : \mathbb{N} \to [0, 1]$ is *negligible* if $\delta(k) < k^{-d}$ for any integer $d > 0$ and sufficiently large $k \in \mathbb{N}$.

Definition 2.1.1 *We say that two distribution ensembles X and Y are* equally distributed *(and write $X \equiv Y$) if for all k and all a we have that $X(k, a) \equiv Y(k, a)$.*

Definition 2.1.2 *Let $\delta : \mathbb{N} \to [0, 1]$. Two distribution ensembles X and Y have* statistical distance δ *if for all sufficiently large k and all a, we have $\mathcal{SD}(X(k, a), Y(k, a)) < \delta(k)$. If δ is negligible, then we say that X and Y are* statistically indistinguishable *(and write $X \approx Y$).*

2.2 Cryptographic Notions

2.2.1 Locally Decodable Code

Definition 2.2.1 *Let k, n and N be positive integers, and $0 < \delta, \epsilon < 1$. A code $\mathbf{C} : \Sigma^n \to \Gamma^N$ is said to be* (k, δ, ϵ)-locally decodable *if there is a probabilistic decoding algorithm $\mathbf{D}$ such that*

- *For every $x \in \Sigma^n$, $i \in [n]$, and $y \in \Gamma^N$ such that $d_H(y, \mathbf{C}(x)) \leq \delta N$, we have*

$$\Pr[\mathbf{D}^y(i) = x_i] \geq 1 - \epsilon,$$

  where the probability is taken over the internal coin tosses of $\mathbf{D}$.

- *In every invocation, $\mathbf{D}$ makes at most k queries.*

The algorithm $\mathbf{D}$ is called a (k, δ, ϵ)-*local decoding algorithm* for $\mathbf{C}$. Parameters k and N are called the *query complexity* and *length* of $\mathbf{C}$, respectively. The *alphabets* Σ and Γ are often taken to be a finite field $\mathbb{F}_q$, where q is a prime power. A k-query LDC $\mathbf{C} : \mathbb{F}_q^n \to \mathbb{F}_q^N$ is *linear* if it is a linear transformation, and *nonadaptive* if $\mathbf{D}$ makes all queries simultaneously in every invocation.

2.2.2 Private Information Retrieval

Definition 2.2.2 *A one-round k-server private information retrieval protocol involves k servers $\mathcal{S}_1, \ldots, \mathcal{S}_k$ who have an n-bit string $x \in \{0,1\}^n$ and a user $\mathcal{U}$ who has an index $i \in [n]$. It is a triplet of algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$. At the beginning of the protocol, $\mathcal{U}$ picks a random string $\mathtt{aux}$, computes a k-tuple of queries $\mathtt{que} = (\mathtt{que}_1, \ldots, \mathtt{que}_k) = \mathcal{Q}(k, n, i, \mathtt{aux})$ and sends each query $\mathtt{que}_j$ to server $\mathcal{S}_j$. After receiving $\mathtt{que}_j$, the server $\mathcal{S}_j$ replies with $\mathtt{ans}_j = \mathcal{A}(k, n, j, x, \mathtt{que}_j)$. At last, $\mathcal{U}$ outputs $\mathcal{C}(k, n, i, \mathtt{aux}, \mathtt{ans}_1, \ldots, \mathtt{ans}_k)$. The protocol as above should satisfy the following requirements:*

Correctness: *For any n, $x \in \{0,1\}^n$, $i \in [n]$, and $\mathtt{aux}$,*

$$\mathcal{C}(k, n, i, \mathtt{aux}, \mathtt{ans}_1, \ldots, \mathtt{ans}_k) = x_i.$$

User Privacy: *For any $i_1, i_2 \in [n]$, $j \in [k]$, and query $\mathtt{que}$,*

$$\Pr[\mathcal{Q}_j(k, n, i_1, \mathtt{aux}) = \mathtt{que}] = \Pr[\mathcal{Q}_j(k, n, i_2, \mathtt{aux}) = \mathtt{que}].$$

The *communication complexity* of $\mathcal{P}$ is the total number of bits exchanged between the user and all servers, maximized over $x \in \{0,1\}^n$, $i \in [n]$, and random string $\mathtt{aux}$.

2.2.3 Extended Private Information Retrieval

A single-database EPIR is a primitive between a database $\mathcal{DB}$ who has N blocks $(R_1, \ldots, R_N) \in (\{0,1\}^{l_1})^N$ and a user $\mathcal{U}$ who wants to evaluate $F(R_i)$ for a function $F \in \mathcal{F}$ and an index $i \in [N]$, where $\mathcal{F}$ is a set of functions from $\{0,1\}^{l_1}$ to $\{0,1\}^*$ and public. It allows $\mathcal{U}$ to learn $F(R_i)$ but no more information on the database blocks while $\mathcal{DB}$ learns no information on (F, i).

We denote by $\mathbf{retrieve}(F, i)$ the query made by a user with input $(F, i) \in \mathcal{F} \times [N]$. Without further notice, algorithms are assumed to be polynomial-time. If an algorithm $\mathcal{A}$ runs in a number of stages, then we write $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \cdots)$. The security is evaluated by an experiment between an adversary and a challenger, where the challenger simulates the protocol executions and answers the adversary's oracle queries. For $\mathcal{A}$ a probabilistic algorithm, we denote by $\mathcal{A}(\mathcal{O}, \mathbf{retrieve})$ the action to run $\mathcal{A}$ with access to any polynomial number of **retrieve** queries generated or answered (depending on the position of the adversary) by the oracle $\mathcal{O}$.

Correctness. An EPIR protocol is said to be *correct* if any query $\mathbf{retrieve}(F, i)$ returns the correct value of $F(R_i)$ except with a negligible probability when $\mathcal{U}$ and $\mathcal{DB}$ follow the protocol specification.

User Privacy. Informally, an EPIR protocol is said to *respect user privacy* if for any query $\mathbf{retrieve}(F, i)$, $\mathcal{DB}$ learns no information on (F, i). Formally, an EPIR protocol is said to respect user privacy if any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$, acting as a malicious database, has only a negligible advantage $|\Pr[b' = b] - \frac{1}{2}|$ in the experiment depicted by Figure 2-1.

In this experiment, the adversary $\mathcal{A}$ is a malicious database $\mathcal{DB}$. The challenger and the adversary proceed as follows:

$\mathbf{Exp}_{\mathcal{A}}^{\text{user-privacy}}$

$$
\begin{array}{rcl}
(R_1, \ldots, R_N) & \leftarrow & \mathcal{A}_1(1^l) \\
1 \leq i_0, i_1 \leq N; F_0, F_1 \in \mathcal{F} & \leftarrow & \mathcal{A}_2(Challenger; \mathbf{retrieve}) \\
b & \leftarrow & \{0,1\} \\
\emptyset & \leftarrow & \mathcal{A}_3(Challenger; \mathbf{retrieve}(F_b, i_b)) \\
b' & \leftarrow & \mathcal{A}_4(Challenger; \mathbf{retrieve})
\end{array}
$$

Figure 2-1: User privacy

1. The adversary $\mathcal{A}_1$ generates N blocks $R = (R_1, R_2, \ldots, R_N)$.

2. The adversary $\mathcal{A}_2$ can request the challenger to start any (polynomial) number of **retrieve** queries. At some point, $\mathcal{A}_2$ outputs (i_0, i_1, F_0, F_1) for a challenge.

3. The challenger randomly chooses $b \in \{0,1\}$ and issues a **retrieve**(F_b, i_b) query to the adversary $\mathcal{A}_3$.

4. The adversary $\mathcal{A}_4$ can continue requesting the challenger to start any (polynomial) number of **retrieve** queries. At some point, $\mathcal{A}_4$ outputs a guess b'.

Database Privacy. Informally, an EPIR protocol is said to *respect database privacy* if a malicious user $\mathcal{U}$ cannot learn more information than $F'(R_{i'})$ for some $(F', i') \in \mathcal{F} \times [N]$ via a query **retrieve**. This intuitive description can be formalized via simulation approach by saying that the user $\mathcal{U}$ cannot determine whether he is interacting with a simulator which takes only $(i', F'(R_{i'}))$ as input, or with $\mathcal{DB}$. We denote by $\mathcal{S}_0$ the database $\mathcal{DB}$. Formally, an EPIR protocol is said to respect database privacy if there is a simulator $\mathcal{S}_1$, which receives an auxiliary input $(i', F'(R_{i'}))$ from a *hypothetical oracle* $\mathcal{O}$ for every query **retrieve**, such that any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, acting as a malicious user, has only a negligible advantage $|\Pr[b' = b] - \frac{1}{2}|$ in the experiment depicted by Figure 2-2.

In this experiment, the adversary $\mathcal{A}$ is a malicious user $\mathcal{U}$. The challenger and the adversary proceed as follows:

$$
\begin{array}{l}
\mathbf{Exp}_{\mathcal{A}}^{\text{database-privacy}} \\
\left|\begin{array}{rcl}
b & \leftarrow & \{0,1\} \\
(R_1,\ldots,R_N) & \leftarrow & \mathcal{A}_1(1^l) \\
b' & \leftarrow & \mathcal{A}_2(\mathcal{S}_b;\textbf{retrieve})
\end{array}\right.
\end{array}
$$

Figure 2-2: Database privacy

1. The challenger randomly chooses $b \in \{0,1\}$. If $b = 0$ then $\mathcal{S}_0$ answers the retrieve queries from the adversary; otherwise $\mathcal{S}_1$ answers such queries.

2. The adversary $\mathcal{A}_1$ generates N blocks $R = (R_1, R_2, \ldots, R_N)$.

3. The adversary $\mathcal{A}_2$ can start any (polynomial) number of **retrieve** queries. At some point, $\mathcal{A}_2$ outputs a guess b'.

The hypothetical oracle $\mathcal{O}$ is assumed to have unlimited computing resources, and $\mathcal{S}_1$ always learns exactly the input related to the request made by the adversary.

2.2.4 Distributed Oblivious Transfer

In this section, we give both informal and formal definitions of one-round (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ protocols.

Definition 2.2.3 (Informal) *Let t, τ, k, l, and n be integers such that $0 \le t, \tau \le k \le l$. A* one-round (t,τ)-private (k,l)-*DOT-*$\binom{n}{1}$ protocol $\mathcal{P}$ *involves:*

- *a* sender $\mathcal{D}$ *who has n secrets $W = W_1, \ldots, W_n$, which are typically n elements of a finite field $\mathbb{F}$;*
- *a* receiver $\mathcal{U}$ *who has an index $i \in [n]$ and wants to retrieve W_i;*
- *additional l* servers $\mathcal{S}_1, \ldots, \mathcal{S}_l$*;*

and consists of two stages:

- *a* setup stage *when $\mathcal{D}$ sends to the server $\mathcal{S}_h$ a message $D_h = \mathcal{D}(h, W, X)$ in a* secure *way for every $h \in [l]$, where X is the random input of $\mathcal{D}$;*

- *a* transfer stage *when* $\mathcal{U}$ *interacts with* k *out of the* l *servers, say* $\mathcal{S}_K = \{\mathcal{S}_h : h \in K\}$ *where* $K \subseteq [l]$ *and* $|K| = k$*, sends to each server* $\mathcal{S}_h$ *a query* $Q_h = \mathcal{Q}(n, i, K, Y)_h$ *and receives the answer* $A_h = \mathcal{S}_h(D_h, Q_h)$*, where* Y *is the random input of* $\mathcal{U}$*;*

such that the following requirements are satisfied:

- Correctness: *The receiver* $\mathcal{U}$ *is able to correctly compute* W_i *after the transfer stage;*
- Receiver's privacy: *A coalition of up to* t *servers learns no information on* i *after the transfer stage;*
- Sender's privacy I: *A coalition of* $\mathcal{U}$ *and up to* τ *servers learns no information on* W *before the transfer stage;*
- Sender's privacy II: *Given the transcript of communication with* k *servers, a coalition of* $\mathcal{U}$ *and any* τ *malicious servers learns at most one secret.*

In the remaining of this section, we shall give the formal definition, which follows [11] and has the strongest requirement for sender's privacy. We denote by $\bar{\mathcal{U}}$ and $\bar{\mathcal{S}}_h$ the corrupted user $\mathcal{U}$ and server $\mathcal{S}_h$, respectively. The following random variables are involved in our formal definition:

- $\mathbf{W}, \mathbf{X}, \mathbf{I}, \mathbf{Y}$: the n secrets of $\mathcal{D}$, the random input of $\mathcal{D}$, the index of $\mathcal{U}$ and the random input of $\mathcal{U}$, respectively;
- $\mathbf{D}_h$, $\mathbf{Q}_h$, $\mathbf{A}_h$: the message sent to server $\mathcal{S}_h$ by $\mathcal{D}$, the query sent to server $\mathcal{S}_h$ by $\mathcal{U}$ and the answer sent to $\mathcal{U}$ by the server $\mathcal{S}_h$, respectively;
- $\mathbf{C}_h$: the transcript of communication between $\mathcal{U}$ and $\mathcal{S}_h$, i.e., $\mathbf{C}_h = (\mathbf{Q}_h, \mathbf{A}_h)$;
- $\bar{\mathbf{Q}}_h, \bar{\mathbf{A}}_h$: the query sent to server $\mathcal{S}_h$ by $\bar{\mathcal{U}}$ and the answer sent to $\mathcal{U}$ by $\bar{\mathcal{S}}_h$ or sent to $\bar{\mathcal{U}}$ by $\mathcal{S}_h$, respectively;
- $\bar{\mathbf{C}}_h$: the transcript of communication between $\bar{\mathcal{U}}$ and $\mathcal{S}_h$, i.e., $\bar{\mathbf{C}}_h = (\bar{\mathbf{Q}}_h, \bar{\mathbf{A}}_h)$.

Let $\mathbf{V}_s$ be a random variable for every integers $s > 0$. For a set of positive integers S, we denote by $\mathbf{V}_S$ the sequence of random variables $\mathbf{V}_s$ indexed by $s \in S$. Let $H(\mathbf{V}) = -\sum_v \Pr[\mathbf{V} = v] \log \Pr[\mathbf{V} = v]$ be the *binary entropy* of $\mathbf{V}$.

Assumptions: As in [73, 11], our formal definition depends on the following assumptions:

- The random input of $\mathcal{U}$ is truly random and independent of the inputs of $\mathcal{D}$:

$$H(\mathbf{Y}|\mathbf{W},\mathbf{X}) = H(\mathbf{Y}); \tag{2.1}$$

- The random input of $\mathcal{D}$ is truly random and independent of the inputs of $\mathcal{U}$:

$$H(\mathbf{X}|\mathbf{I},\mathbf{Y}) = H(\mathbf{X});$$

- The private input of $\mathcal{U}$ is independent of any other data in the setup stage:

$$H(\mathbf{I}) = H(\mathbf{I}|\mathbf{W},\mathbf{X},\mathbf{D}_{[l]},\mathbf{Y}); \tag{2.2}$$

- Every index $i \in [n]$ will be the choice of $\mathcal{U}$ with positive probability:

$$\Pr[\mathbf{I} = i] > 0 \text{ for every } i \in [n];$$

- The secrets are totally independent, i.e., for every $K, K' \subseteq [n]$ s.t. $K \cap K' = \emptyset$:

$$H(\mathbf{W}_{K'}|\mathbf{W}_K) = H(\mathbf{W}_{K'}); \tag{2.3}$$

- For $K \subseteq [l]$, the messages sent to $\mathcal{S}_K$ are determined by the inputs of $\mathcal{D}$:

$$H(\mathbf{D}_K|\mathbf{W},\mathbf{X}) = 0; \tag{2.4}$$

- For $K \subseteq [l]$, the queries sent to $\mathcal{S}_K$ are determined by the inputs of $\mathcal{U}$:

$$H(\mathbf{Q}_K|\mathbf{I},\mathbf{Y}) = 0; \tag{2.5}$$

- For $K \subseteq [l]$, the answers of $\mathcal{S}_K$ are determined by the information it received:

$$H(\mathbf{A}_K|\mathbf{Q}_K,\mathbf{D}_K) = 0. \tag{2.6}$$

Remark 1: The sender's secrets are assumed to be *implication-free* in [11], i.e., $H(\mathbf{W}_i|\mathbf{W}_j) > 0$ whenever $i \neq j$. However, the secrets are usually independent. Hence, we assume that (2.3) holds in our model.

Remark 2: Following [11], we suppose that the message $\mathbf{D}_h$ contains the randomness needed by $\mathcal{S}_h$ for every $h \in [l]$. Hence, we can assume that the servers are deterministic in our model. In particular, (2.6) holds.

We are ready to *formalize* the correctness and privacy requirements which have been stated in the informal definition.

- CORRECTNESS: The protocol $\mathcal{P}$ is *correct* if for every $i \in [n]$ and every k-subset $K \subseteq [l]$, the following identity holds:

$$H(\mathbf{W}_i|\mathbf{I} = i, \mathbf{Y}, \mathbf{C}_K) = 0; \tag{2.7}$$

- RECEIVER'S PRIVACY: For every $T \subseteq [l]$ such that $|T| \leq t$,

$$H(\mathbf{I}|\mathbf{D}_T, \mathbf{Q}_T) = H(\mathbf{I});$$

- SENDER'S PRIVACY I: For every $T \subseteq [l]$ such that $|T| \leq \tau$,

$$H(\mathbf{W}|\mathbf{I}, \mathbf{Y}, \mathbf{D}_T) = H(\mathbf{W});$$

- SENDER'S PRIVACY II: For every $K, T \subseteq [l]$ such that $|K| = k, |T| \leq \tau$, index i and random input Y,

$$H(\mathbf{W}|\mathbf{I} = i, \mathbf{Y} = Y, \bar{\mathbf{C}}_K, \mathbf{D}_T, \mathbf{W}_i) = H(\mathbf{W}|\mathbf{W}_i). \tag{2.8}$$

Remark 3: We are studying unconditionally secure protocols. Therefore, following [11], we can suppose that the corrupted receiver is deterministic. In other words, we suppose that i is the input *actually contributed* by $\bar{\mathcal{U}}$ and Y is the randomness *actually used* by $\bar{\mathcal{U}}$. If $i \notin [n]$, (2.8) should be interpreted as $H(\mathbf{W}|\mathbf{I} = i, \mathbf{Y} = Y, \bar{\mathbf{C}}_K, \mathbf{D}_T) = H(\mathbf{W})$.

Remark 4: The $(k-1, 0)$-private (k, l)-DOT-$\binom{n}{1}$ have been called *strongly* private in [11] and are necessarily multi-round. We obtain one-round (t, τ)-private (k, l)-DOT-$\binom{n}{1}$ for a variety of parameters $(t, \tau) \neq (k-1, 0)$ in Chapter 5.

2.2.5 Oblivious Linear Function Evaluation

In this section, we formally define the cryptographic notions we shall study in Chapter 6, namely, $\binom{n}{1}$-OT and $\mathcal{C}$-OLFE$_n$. First of all, their functionalities are defined below.

Definition 2.2.4 $\binom{n}{1}$-OT *is a primitive between a sender Alice who has as input n secrets $s = (s_1, \ldots, s_n)$ and no output, and a receiver Bob who has as input a choice $c \in [n]$ and outputs s_c.*

Definition 2.2.5 *Let $\mathbb{F}$ be a finite field and $\mathcal{C} \subseteq \mathbb{F}^n$. An* oblivious n-variate linear function evaluation with choice space $\mathcal{C}$ *(denoted by $\mathcal{C}$-OLFE$_n$) is a primitive between a sender Alice who has as input $s(x) = \sum_{i=1}^{n} s_i x_i \in \mathbb{F}[x]$ (equivalently, Alice's input can be taken to be $s = (s_1, \ldots, s_n)$ as well) and no output, and a receiver Bob who has as input a choice $c \in \mathcal{C}$ and outputs $s(c)$.*

The security requirements on $\binom{n}{1}$-OT and $\mathcal{C}$-OLFE$_n$ are defined in terms of general secure two-party function evaluation [19]. We follow the security definitions of [19] because both primitives are often used as subprotocols and residue in very complicated external environments. Let $f(k, x_1, x_2)$ be a *two-party function* which maps any tuple (k, x_1, x_2) of security parameter k, inputs x_1 and x_2 to a pair of outputs. A *two-party protocol* for f is a pair $(\mathcal{P}_1, \mathcal{P}_2)$ of interactive Turing machines (ref. [47]), where each machine $\mathcal{P}_i$ starts with a pair (k, x_i). We define the security of a two-party protocol by comparing the ideal model and hybrid model. In the *ideal model*, the two parties do not interact with each other and evaluate $f(k, x_1, x_2)$ with the help of a trusted third party $\mathcal{T}^f$ (which implements the functionality of f) and in the presence of an ideal model adversary $\mathcal{S}$. Let g be a two-party function as well. In the *g-hybrid model*, the two parties interact with each other and evaluate $f(k, x_1, x_2)$ with the help of a trusted third party $\mathcal{T}^g$ (which implements the functionality of g) and in the presence of a g-hybrid model adversary $\mathcal{H}$.

Ideal model. Let $\mathcal{P}_i$ be the party corrupted by $\mathcal{S}$ and z be the auxiliary input obtained by $\mathcal{S}$, respectively. Then $\mathcal{P}_{3-i}$ is not corrupted. The entities $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{T}^f$ and $\mathcal{S}$ start with $(k, x_1), (k, x_2), k$ and (k, x_i, z), respectively.

- SUBSTITUTION: *$\mathcal{S}$ instructs $\mathcal{P}_i$ to substitute its input x_i with an arbitrary x_i'.*
- COMPUTATION: *The parties $\mathcal{P}_{3-i}$ and $\mathcal{P}_i$ send x_{3-i} and x_i' to $\mathcal{T}^f$, respectively; then $\mathcal{T}^f$ computes $(f_1, f_2) = f(k, x_1', x_2)$ when $i = 1$ and $(f_1, f_2) = f(k, x_1, x_2')$ when $i = 2$, and sends f_1 and f_2 to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively.*
- OUTPUT: *$\mathcal{P}_{3-i}$ outputs f_{3-i}, $\mathcal{P}_i$ outputs $\perp$ (which indicates that it is corrupted) and $\mathcal{S}$ outputs an arbitrary function of its view of the computation.*

The *global output* $\mathrm{IDEAL}_{f,\mathcal{S}}(k, \boldsymbol{x}, z)$ of the ideal model is defined to be the concatenation of the outputs of $\mathcal{S}$ and $\mathcal{P}_1, \mathcal{P}_2$, where $\boldsymbol{x} = x_1 x_2$. We denote $\mathrm{IDEAL}_{f,\mathcal{S}} = \{\mathrm{IDEAL}_{f,\mathcal{S}}(k, \boldsymbol{x}, z)\}_{k \in \mathbb{N}, \langle \boldsymbol{x}, z \rangle \in \{0,1\}^*}$.

Hybrid model. Let $\mathcal{P}_i$ be the party corrupted by $\mathcal{H}$ and z be the auxiliary input obtained by $\mathcal{S}$, respectively. Then $\mathcal{P}_{3-i}$ is not corrupted. The entities $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{T}^g$ and $\mathcal{H}$ start with $(k, x_1), (k, x_2), k$ and (k, x_i, z), respectively.

- COMPUTATION: *The computation consists of a number of* interactive rounds *when the two parties directly exchange messages with each other and* g-rounds *when the two parties do not communicate with each other but invoke $\mathcal{T}^g$ in order to evaluate g.*
  - *In each interactive round, only one party is active. If $\mathcal{P}_{3-i}$ is active, then it sends a message m_{3-i} to $\mathcal{P}_i$ according to the protocol specification. If $\mathcal{P}_i$ is active, then the adversary $\mathcal{H}$ generates a message m_i and instructs $\mathcal{P}_i$ to send m_i to $\mathcal{P}_{3-i}$.*
  - *In each g-round, both parties are active. $\mathcal{P}_{3-i}$ sends an input y_{3-i} to $\mathcal{T}^g$ which is specified by the protocol; $\mathcal{H}$ decides an input y_i and instructs $\mathcal{P}_i$ to send y_i to $\mathcal{T}^g$. At last, $\mathcal{T}^g$ evaluates $(g_1, g_2) = g(k, y_1, y_2)$ and sends g_1 and g_2 to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively.*
- OUTPUT: *$\mathcal{P}_{3-i}$ outputs whatever specified by the protocol; $\mathcal{P}_i$ outputs $\perp$ and $\mathcal{H}$ outputs an arbitrary function of its* view *of the computation.*

The *global output* $\mathrm{EXEC}_{\pi^g,\mathcal{H}}(k, \boldsymbol{x}, z)$ of the hybrid model is defined to be the concatenation of the outputs of $\mathcal{H}$ and $\mathcal{P}_1, \mathcal{P}_2$, where $\boldsymbol{x} = x_1 x_2$. We denote $\mathrm{EXEC}_{\pi^g,\mathcal{H}} = \{\mathrm{EXEC}_{\pi^g,\mathcal{H}}(k, \boldsymbol{x}, z)\}_{k \in \mathbb{N}, \langle \boldsymbol{x}, z \rangle \in \{0,1\}^*}$.

Real-ideal world approach: One can similarly define the *real model* as we have done for the ideal model and hybrid model. The real model is almost identical to the hybrid model except the computation is done without the help of $\mathcal{T}^g$. In other words, the two parties communicate with each other directly while the behavior of one party is controlled by the adversary. We emphasize the hybrid model instead of real model because our construction in Chapter 6 involves the trusted third party $\mathcal{T}^g$. More precisely, in our construction, the f is $\binom{n}{1}$-OT and the g is the newly defined $\mathcal{C}$-OLFE$_n$. A protocol π^g for f in the g-hybrid model *securely evaluates* f if it emulates the ideal model for f, in the sense that any effect on the g-hybrid model computation achieved by $\mathcal{H}$ can be achieved in the ideal model computation by some ideal model adversary $\mathcal{S}$. Formally, we have the following definition.

Definition 2.2.6 *Let f and g be two-party functions. A two-party protocol π^g for f in the g-hybrid model* evaluates f statistically securely *if for every g-hybrid model adversary $\mathcal{H}$, there is an ideal model adversary $\mathcal{S}$ whose running time is polynomial in that of $\mathcal{H}$ such that* $\text{IDEAL}_{f,\mathcal{S}} \approx \text{EXEC}_{\pi^g,\mathcal{H}}$. *In particular, if* $\text{IDEAL}_{f,\mathcal{S}} \equiv \text{EXEC}_{\pi^g,\mathcal{H}}$, *then we say that π^g* evaluates f perfectly securely.

A *reduction* from f to g is a two-party protocol π^g for f in the g-hybrid model. We say that a reduction π^g from f to g *implements f statistically (resp. perfectly) securely* if π^g evaluates f statistically (resp. perfectly) securely in the g-hybrid model.

2.2.6 ElGamal Encryption Scheme

Let $K = \mathbb{F}_p, L = \mathbb{F}_q$ and $\mathbb{G} = L^*$, where p is a prime and $q = p^n$ for an integer $n \geq 2$. Let g be a generator of $\mathbb{G}$. The *ElGamal encryption scheme* over $\mathbb{G}$ is a triplet of algorithms $\Pi = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$, where

- **Gen** is a key generation algorithm which takes as input a security parameter 1^k and proceeds as follows:
  - generates the parameters p, n, q and g;
  - picks $x \leftarrow \mathbb{Z}_q$ randomly and computes $y = g^x$;

– outputs $pk = (q, g, y)$ as the public key and $sk = x$ as the secret key.

- **Enc** is an encryption algorithm which takes as input a message $m \in \mathbb{G}$, the public key pk and proceeds as follows
  - picks $r \leftarrow \mathbb{Z}_q$ randomly;
  - outputs $c = (g^r, y^r m)$ as the ciphertext.
- **Dec** is a decryption algorithm which takes as input the ciphertext $c = (c_1, c_2) \in \mathbb{G}^2$, the secret key $sk = x$ and outputs $c_2 \cdot c_1^{-x}$ as the message.

2.2.7 Secret Sharing

A secret sharing scheme Π [84, 10] is a method of distributing a *secret* s among n participants $\mathcal{P}_1, \ldots, \mathcal{P}_n$ by a dealer $\mathcal{D}$ such that any authorized subset $A \subseteq \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ of participants can recover s by pooling their *shares* while any unauthorized subset of participants can learn even no partial information about s. The set Γ of all authorized subsets A is called the *access structure* of the secret sharing scheme. Clearly, Γ must be *monotone increasing*, i.e., $A \in \Gamma, A \subseteq B \Rightarrow B \in \Gamma$. A secret sharing scheme Π *realizes* an access structure Γ if the set of all authorized subsets of participants in Π is equal to Γ. Given an access structure Γ, there may be many secret sharing schemes realizing it. The earliest secret sharing schemes by [84, 10] realize the (t, n)-*threshold access structure*, where Γ is the set of all subsets of $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ of cardinality at least t, where $t \leq n$ is called the *threshold*.

Let Γ be any access structure. Let $\mathbf{S}$ be the probability distribution of the secret s and $\mathbf{A}$ be the joint probability distribution of the shares obtained by participants in $A \subseteq \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$. A secret sharing scheme realizing Γ can be formally defined as follows.

Definition 2.2.7 *A secret sharing scheme* Π realizing *an access structure* Γ *is an algorithm for assigning shares of a secret* s *to each participant such that any authorized subset of participants can jointly determine* s*. More formally,* Π *can be defined in terms of two algorithms:*

- *the share distribution algorithm* $\texttt{Gen}(s, r)$ *which takes as input a secret* s *and randomness* r *and outputs a list* $(s_1, \ldots, s_n)$ *of shares that are distributed to the* n *participants;*

- *the reconstruction algorithm* `Rec` *which takes as input a set of valid shares held by an authorized subset* $A \in \Gamma$ *and outputs the secret* s*. In other words, the shares held by* $A \in \Gamma$ *should satisfy the following requirement:* $H(\mathbf{S}|\mathbf{A}) = 0$.

The secret sharing scheme Π *is said to be* perfect *if any unauthorized subset of participants learns absolutely no information on* s*, i.e.,* $H(\mathbf{S}|\mathbf{A}) = H(\mathbf{S})$ *for every* $A \notin \Gamma$.

In Chapter 5, we shall use the *replication-based threshold secret sharing scheme* $\text{CNF}_{\tau,k}$ introduced by [54]. Formally, the $\text{CNF}_{\tau,k}$ involves a dealer $\mathcal{D}$ who has a secret $g \in \mathbb{G}$ and k participants $\mathcal{S}_1, \ldots, \mathcal{S}_k$, where $\mathbb{G}$ is an abelian group. The share distribution algorithm of the scheme proceeds as follows:

- The dealer picks $\binom{k}{\tau}$ random group elements, each of which is indexed by a τ-subset of $[k]$, say $\{g_T : T \subseteq [k], |T| = \tau\}$, such that $\sum_T g_T = g$;

- The dealer sends $\{g_T : h \notin T\}$ to $\mathcal{S}_h$ for every $h \in [k]$.

Clearly, any $\tau + 1$ or more participants know all the shares and are able to recover the secret g. On the other hand, any τ or less participants miss at least one of the $\binom{k}{\tau}$ shares and can learn absolutely no information about g. The $\text{CNF}_{\tau,k}$ actually realizes the $(\tau + 1, k)$-threshold access structure.

Chapter 3

On Bringer-Chabanne EPIR Protocol for Polynomial Evaluation

In this chapter, we show that the Bringer-Chabanne EPIR protocol for polynomial evaluation does not satisfy the correctness requirement (defined in Section 2.2.3) as it has been claimed by [15]. In particular, we show that the protocol does not give the user the expected result with large probability when the polynomial of the user has some special property. Furthermore, the class of the polynomials having this special property is infinite. Our argument is by contradiction. Firstly, we give a restricted version of the protocol in which the database is deterministic and only has one block (i.e., $N = 1$). Secondly, we show that the restricted version does not satisfy the correctness requirement when the user wants to evaluate any polynomial with the special property.

3.1 Bringer-Chabanne EPIR Protocol

The EPIR protocols for testing equality and computing weighted Hamming distance of [16] are based on a pre-processing technique. Specifically, the user sends an encryption of its input (F, i) to $\mathcal{DB}$, who then computes a temporary database which contains an encryption of $F(R_i)$. Finally, the user executes a single-database CPIR protocol with $\mathcal{DB}$ to retrieve the encryption of $F(R_i)$. This technique does not allow the evaluation

1. $\mathcal{U}$: Generates an ElGamal key pair $(pk, sk) = ((q, g, y), x)$, where $q = p^n$ for a prime p and an integer $n > 0$, g is a generator of $\mathbb{F}_q^*$, and $y = g^x$ for a randomly chosen integer $x \in \mathbb{Z}_q$. As in Section 2.2.6, denote $K = \mathbb{F}_p$, $L = \mathbb{F}_q$ and $\mathbb{G} = L^*$. The user sends pk to $\mathcal{DB}$ such that $\mathcal{DB}$ can verify the validity of pk. In practice, the validity of pk can be certified by a TTP (Trusted Third Party), and the same pk can be used by the user for all his queries. (Clearly, there are polynomials $G(t), Y(t) \in \mathbb{F}_p[t]$ of degree $< n$ such that $g = G(\alpha)$ and $y = Y(\alpha)$, where $\alpha \in \mathbb{F}_q$ is a primitive element.)

2. $\mathcal{U}$: For any polynomial function $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ and any index $1 \leq i \leq N$, computes $C_1, \ldots, C_N$ and sends them to $\mathcal{DB}$ where

   - $C_i = \mathbf{Enc}(F(\alpha) + r) = (G(\alpha)^{r_i}, Y(\alpha)^{r_i}(F(\alpha) + r))$
   - and $C_j = \mathbf{Enc}(1) = (G(\alpha)^{r_j}, Y(\alpha)^{r_j})$ for all $j \neq i$,

   with randomly chosen $r \in \mathbb{F}_p, r_j \in \mathbb{Z}_q (1 \leq j \leq N)$. Each C_j can be written as $C_j = (V_j(\alpha), W_j(\alpha))$, where $V_j(t), W_j(t) \in \mathbb{F}_p[t]$ are of degree $< n$.

3. $\mathcal{DB}$: After receiving the C_j , checks that they are nontrivial ElGamal ciphertexts and computes $C_j(R_j) = (V_j(R_j), W_j(R_j))$ by replacing each occurrence of α (resp. α^l for all power $l < n$) with R_j (resp. with R_j^l).

4. $\mathcal{DB}$: Performs the product of all the C_j together with a random encryption of 1, $\mathbf{Enc}(1) = (g^{r'}, y^{r'})$, to obtain

$$\mathbf{Enc}(1) \times \prod_{j=1}^{N} C_j(R_j) = \left(g^{r'} \prod_{j=1}^{N} G(R_j)^{r_j}, y^{r'} \left(\prod_{j=1}^{N} Y(R_j)^{r_j} \right) (F(R_i) + r) \right).$$

   Sends the above information to $\mathcal{U}$.

5. $\mathcal{U}$: Outputs $\mathbf{Dec}(sk, \mathbf{Enc}(1) \prod_{j=1}^{N} C_j(R_j)) - r$ as $F(R_i)$.

Figure 3-1: Bringer-Chabanne EPIR protocol

of generic functions and incurs heavy computation during the generation of the temporary database. The Bringer-Chabanne EPIR protocol aims at avoiding these deficiencies. It is based on ElGamal encryption schemes (defined in Section 2.2.6) over the multiplicative groups of finite fields.

Figure 3-1 is the Bringer-Chabanne EPIR protocol. The authors of the protocol expect to embed the description of the polynomial $F(t) \in L[t]$ chosen by $\mathcal{U}$ into an ElGamal ciphertext such that it can be evaluated by $\mathcal{DB}$ in an oblivious way. At the beginning of the protocol, the user set up an ElGamal encryption scheme and gives his public key to the database. Then he sends N ElGamal ciphertexts $C_1, \ldots, C_N$ to the database, where the C_j is a random encryption of 1 whenever $j \neq i$ and C_i is a random encryption of $F(\alpha) + r$, where F is the user's polynomial, α is the primitive element of the finite field and r is a random element of the prime field. The database thinks of each ElGamal ciphertext C_j as a pair of polynomials (V_j, W_j) in α. It evaluates the polynomials (V_j, W_j) on R_j for every $j \in [N]$ and then sends a random encryption of $\prod_{j=1}^{N} C_j$ to the user. At last, the user decrypt the message given by $\mathcal{DB}$ in order to learn $F(R_i)$. Clearly, $C_j = (V_j(\alpha), W_j(\alpha))$ is a valid ElGamal encryption under the secret key chosen by $\mathcal{U}$ for every $j \in [N]$. The construction would be correct if $(V_j(R_j), W_j(R_j))$ is also a valid ElGamal ciphertext under the same secret key for every $j \in [N]$. Unfortunately, this is not the case in general! That's why the EPIR protocol may not give $\mathcal{U}$ the correct output.

Following the notations in Figure 3-1, Bringer et al. [15] required that for every $j \in [N]$, the database block R_j should belong to $\mathbb{D}$, where

$$\mathbb{D} = \{\beta \in \mathbb{G} : Y(\beta) = G(\beta)^x \text{ and } G(\beta) \neq 0\}.$$

Under this requirement, Bringer et al. [15] claimed that $\mathcal{U}$ can correctly evaluate any polynomial in $\mathcal{F} = L[t]$ using the protocol depicted by Figure 3-1.

Claim 3.1.1 (Section 4.4 of [15]) *A query* (say **retrieve**(F, i)) *gives the expected result* (i.e., $F(R_i)$) *as soon as there is no index j for which one of the values $G(R_j)$ or $Y(R_j)$ is zero (i.e., it is required that $R_j \in \mathbb{D}$ for $j \in [N]$), which may occur only with*

a negligible probability in practice, leading to the correctness of the EPIR protocol.

3.2 Analysis of Bringer-Chabanne EPIR Protocol

In this section, we show that Bringer-Chabanne EPIR protocol does not satisfy the correctness requirement defined in Section 2.2.3. To simplify the argument, we give a restricted version of Bringer-Chabanne EPIR protocol in which $\mathcal{DB}$ is deterministic and $N = 1$. The restricted version satisfies the correctness requirement as long as Bringer-Chabanne EPIR protocol satisfies the correctness requirement.

1. $\mathcal{U}$: Generates an ElGamal key pair $(pk, sk) = ((q, g, y), x)$, where $q = p^n$ for a prime p and an integer $n > 0$, g is a generator of $\mathbb{F}_q^*$, and $y = g^x$ for a randomly chosen integer $x \in \mathbb{Z}_q$. Let $K = \mathbb{F}_p$, $L = \mathbb{F}_q$ and $\mathbb{G} = L^*$. The user sends pk to $\mathcal{DB}$ such that $\mathcal{DB}$ can verify the validity of pk. Let $G(t), Y(t) \in \mathbb{F}_p[t]$ be polynomials of degree $< n$ such that $g = G(\alpha)$ and $y = Y(\alpha)$, where $\alpha \in \mathbb{F}_q$ is a primitive element.
2. $\mathcal{U}$: For any polynomial function $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$, computes $C = \mathbf{Enc}(F(\alpha) + r) = (G(\alpha)^s, Y(\alpha)^s(F(\alpha) + r))$ and sends it to $\mathcal{DB}$ where $r \in \mathbb{F}_p, s \in \mathbb{Z}_q$ are randomly chosen. The ciphertext C can be written as $C = (V(\alpha), W(\alpha))$ where $V(t), W(t) \in K[t]$ are of degree $< n$.
3. $\mathcal{DB}$: After receiving C , checks that it is a nontrivial ElGamal ciphertext and computes $C(R) = (V(R), W(R))$ by replacing each occurrence of α (resp. α^l for all power $l < n$) with R (resp. with R^l).
4. $\mathcal{DB}$: Sends $C(R)$ to $\mathcal{U}$.
5. $\mathcal{U}$: Outputs $\mathbf{Dec}(sk, C(R)) - r$ as $F(R)$.

Figure 3-2: A restricted version of Bringer-Chabanne EPIR protocol

3.2.1 Restricted Version and Counterexample

Restricted version. At step 4 of Bringer-Chabanne EPIR protocol, $\mathcal{DB}$ is randomizing the product $\prod_{j=1}^{N} C_j(R_j)$ and sending $\mathbf{Enc}(1) \cdot \prod_{j=1}^{N} C_j(R_j)$ to the user. We note that the user could have computed the same output if $\mathcal{DB}$ merely sends $\prod_{j=1}^{N} C_j(R_j)$. Therefore, we can safely modify step 4 such that $\mathcal{DB}$ merely sends $\prod_{j=1}^{N} C_j(R_j)$ to $\mathcal{U}$ with no impact on the correctness of the protocol. Let $i = N = 1$. The we have the restricted version (see Figure 3-2). Clearly, if Claim 3.1.1 holds, then we have:

Claim 3.2.1 *A query* (say **retrieve**$(F, 1)$) *in an execution of the restricted version gives* $\mathcal{U}$ *the expected result* (i.e., $F(R)$) *for any* $R \in \mathbb{D}$.

1. $\mathcal{U}$: Let $p = 2, n = 3, K = \mathbb{F}_2, L = \mathbb{F}_{2^3}$ and $\mathbb{G} = L^*$. Let $\alpha = g \in \mathbb{G}$ be a generator of $\mathbb{G}$ with minimal polynomial $\mathrm{Min}_g(t) = t^3 + t + 1 \in K[t]$. Let $(pk, sk) = ((7, g, g^2 + 1), 6)$ be a key pair for the ElGamal encryption scheme over $\mathbb{G}$, where $x = 6$ and $y = g^2 + 1$. The user sends pk to $\mathcal{DB}$ such that $\mathcal{DB}$ can verify the validity of pk. Clearly, $g = G(\alpha)$ and $y = Y(\alpha)$ for polynomials $G(t) = t, Y(t) = t^2 + 1 \in K[t]$ of degree $< n$. The field elements $R \in L$ which satisfy equality $Y(R) = G(R)^x$ are g, g^2 and $g^2 + g$.
2. $\mathcal{U}$: Takes $F(t) = g \in L[t]$, $s = 6 \in \mathbb{Z}_7, r = 1 \in K$ and sends $C = \mathbf{Enc}(F(\alpha) + r) = (G(\alpha)^s, Y(\alpha)^s(F(\alpha) + r)) = (g^2 + 1, g^2 + g)$ to $\mathcal{DB}$. Clearly, we have that $V(t) = t^2 + 1$ and $W(t) = t^2 + t$.
3. $\mathcal{DB}$: Let $R = g^2 + g \in \mathbb{G}$. After receiving C, $\mathcal{DB}$ checks that C is a nontrivial ElGamal ciphertext and computes $C(R) = (V(R), W(R)) = (g + 1, g^2)$.
4. $\mathcal{DB}$: Sends $C(R) = (g + 1, g^2)$ to $\mathcal{U}$.
5. $\mathcal{U}$: Outputs $\mathbf{Dec}(sk, C(R)) - r = g^2 + g$ as $F(R)$, which is absurd.

Figure 3-3: An execution of the restricted version

Counterexample. We show that Claim 3.2.1 does not hold by a counterexample. Let $p = 2, n = 3, K = \mathbb{F}_2, L = \mathbb{F}_{2^3}$ and $\mathbb{G} = L^*$. Let $\alpha = g \in \mathbb{G}$ be a generator of

$\mathbb{G}$ with minimal polynomial $\text{Min}_g(t) = t^3 + t + 1 \in K[t]$. Figure 3-3 depicts an execution of the restricted version which does not give $\mathcal{U}$ the expected result. We shall see why this happens. It is clear that $C = (g^2 + 1, g^2 + g)$ is a valid ElGamal encryption under the secret key $x = 6$. However, as we have pointed out in Section 3.1, $C(R) = (g+1, g^2)$ is no longer a valid ElGamal encryption under the same secret key $x = 6$. Actually, $C(R)$ is a valid ElGamal encryption under the secret key $x = 5$ instead of $x = 6$! That's why this execution gives $\mathcal{U}$ an incorrect output!

3.2.2 Failure Probability

We have seen that the restricted version may not give $\mathcal{U}$ the expected result in the previous section. However, we cannot conclude that the restricted version does not satisfy the correctness requirement defined in Section 2.2.3. In fact, an EPIR protocol is said to be correct as long as it always gives $\mathcal{U}$ the expected result for any input $(F(t), i) \in L[t] \times [n]$ *except with a negligible probability*. Therefore, as a collection of probabilistic algorithms, an EPIR protocol is *allowed to fail with a negligible probability*. Hence, to show that the restricted version does not satisfy the correctness requirement, it is required to compute the failure probability of the protocol, i.e., the probability that the protocol does not give $\mathcal{U}$ the expected result. In this section, we show, through experimental results, that the restricted version does fail with large probability for certain choices of $F(t)$ (e.g., $F(t) = g$).

From now on, we fix $p = 2$ to be the characteristic of all related finite fields. However, we stress that our methodology is applicable to any characteristic p. Following the notations of Section 2.2.6 and Section 3.1, let $K = \mathbb{F}_2$ and $L = \mathbb{F}_{2^n}$ for an integer $n \geq 2$. Let $\mathbb{G} = L^*$ be the multiplicative group of L of order $q = 2^n - 1$ and g be a generator of $\mathbb{G}$. W.l.o.g., we suppose $\alpha = g$. Then $G(t) = t \in K[t]$ is the polynomial of degree less than n such that $G(\alpha) = g$. For every $x \in \mathbb{Z}_q$, let $Y(t) \in K[t]$ be the polynomial of degree less than n such that $Y(\alpha) = y = g^x$. We define

$$D(t) = G(t)^x + Y(t) = t^x + Y(t) \in K[t].$$

Then the set of database blocks which satisfy the requirements in Section 3.1 is

$$\mathbb{D}_{n,g,x} = \{\beta \in \mathbb{G} | D(\beta) = 0\}.$$

We say that an execution of the restricted version is *parameterized* by the parameters (n, g, x, F, s, r, R) if $x \in \mathbb{Z}_q, F(t) \in L[t], s \in \mathbb{Z}_q, r \in K$ and $R \in \mathbb{D}_{n,g,x}$ are the private key, the polynomial to be evaluated, the randomness used at step 2 and the database block of $\mathcal{DB}$, respectively. Let $V(t), W(t) \in K[t]$ be the polynomials of degree less than n such that $V(g) = g^s$ and $W(g) = y^s(F(g)+r)$. Then the execution of the restricted version parameterized by (n, g, x, F, s, r, R) gives $\mathcal{U}$ the expected result if and only if $V(R) \neq 0$ and $E(R) = 0$, where

$$E(t) = W(t) + V(t)^x(F(t) + r). \tag{3.1}$$

For an execution of the restricted version parameterized by (n, g, x, F, s, r, R), we define

$$\mathbf{H}_{x,s,r,F,R} = \begin{cases} 1 & \text{if } V(R) \neq 0 \text{ and } E(R) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then the execution fails if and only if $\mathbf{H}_{x,s,r,F,R} = 0$. Therefore, the probability that an execution of the restricted version fails when $x \in \mathbb{Z}_q$ is the private key and $F(t) \in L[t]$ is the polynomial to be evaluated by $\mathcal{U}$ is equal to

$$\epsilon(n, g, x, F) = \Pr\left[s \leftarrow \mathbb{Z}_q, r \leftarrow K, R \leftarrow \mathbb{D}_{n,g,x} : \mathbf{H}_{x,s,r,F,R} = 0\right].$$

Since s, r and R are uniformly distributed, we have that

$$\epsilon(n, g, x, F) = \frac{\sum_{s \in \mathbb{Z}_q} \sum_{r \in K} \sum_{R \in \mathbb{D}_{n,g,x}} (1 - \mathbf{H}_{x,s,r,F,R})}{2q \cdot |\mathbb{D}_{n,g,x}|}. \tag{3.2}$$

It follows that the probability that the restricted version fails when $F(t) \in L[t]$ is the

polynomial to be evaluated by $\mathcal{U}$ is equal to

$$\eta(n, g, F) = \frac{1}{q} \sum_{x \in \mathbb{Z}_q} \epsilon(n, g, x, F). \tag{3.3}$$

The values of $\eta(n, g, F)$ for $2 \leq n \leq 9$ and $F(t) = g$ are quite large and enumerated in Table 3.1.

n	$\mathrm{Min}_g(t)$	$\eta(n, g, g)$	n	$\mathrm{Min}_g(t)$	$\eta(n, g, g)$
2	$t^2 + t + 1$	0.61111	6	$t^6 + t^4 + t^3 + t + 1$	0.87719
3	$t^3 + t + 1$	0.74271	7	$t^7 + t + 1$	0.87895
4	$t^4 + t + 1$	0.81537	8	$t^8 + t^4 + t^3 + t^2 + 1$	0.89809
5	$t^5 + t^2 + 1$	0.83630	9	$t^9 + t^4 + 1$	0.90358

Table 3.1: Failure probability of the restricted version

3.2.3 Analysis of the Restricted Version

In this section, we show that the restricted version fails with large probability when $F(t) = g$. Specifically, for every integer $n \geq 2$, we give lower bound on $\eta(n, g, g)$.

We follow the notations in Section 2.1.2 and Section 3.2.2. Let U be the set of coset representatives of all cyclotomic cosets of 2 mod q. For any $u \in U$, $\mathbb{E}_u$ is the cyclotomic coset of 2 modulo q with coset representative u. For every integer $d > 0$, we denote by $N_2(d)$ the number of monic irreducible polynomials of degree d in $K[t]$.

Lemma 3.2.1 *The following statements hold:*

1. *For every $u \in U$, the cardinality of $\mathbb{E}_u$ is a divisor of n.*

2. *For every positive integer $d|n$, the number of cyclotomic cosets of 2 mod q of cardinality d is $N_2(d)$.*

3. *([65]) For every integer $d \geq 2$, we have that $N_2(d) \leq \frac{1}{d}(2^d - 2)$.*

For every $u \in U$, we denote by

$$\mathbf{D}_u = \left\{ g^j | j \in \mathbb{E}_u \right\}$$

the set of field elements in L which have the same minimal polynomial over K with g^u. For every $x \in \mathbb{Z}_q$, it is clear that there is a subset $U_x \subseteq U$ of coset representatives such that

$$\mathbb{D}_{n,g,x} = \bigcup_{u \in U_x} \mathbf{D}_u. \tag{3.4}$$

Lemma 3.2.2 *For every $x \in \mathbb{Z}_q$, we have that $1 \in U_x$.*

Proof: It follows from the fact that $D(t) \in K[t]$ and $D(g) = 0$. □

By Equality (3.1), $E(t)$ is determined by the parameters $g \in \mathbb{G}, x \in \mathbb{Z}_q, F(t) \in L[t], s \in \mathbb{Z}_q$ and $r \in K$. Next lemma shows that $E(t)$ and $D(t)$ only share a very small number of roots in L when $F(t) = g$.

Lemma 3.2.3 *Suppose $F(t) = g$. Then for every $x \in \mathbb{Z}_q, u \in U_x, s \in \mathbb{Z}_q$ and $r \in K$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$ or $E(t)$ has at most one root in $\mathbf{D}_u$.*

Proof: If $V(g^u) = 0$, then $V(g^{2^j \cdot u}) = V(g^u)^{2^j} = 0$ for any $j \in \mathbb{N}$, i.e., $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$. Otherwise, we show that $E(t)$ has at most one root in $\mathbf{D}_u$. Due to Equality (3.1), we have that $E(t) = W(t) + V(t)^x(g + r)$. Suppose that $E(t)$ has two different roots in $\mathbf{D}_u$, say $g^{u \cdot 2^j}$ and $g^{u \cdot 2^k}$, where $0 \leq j < k < n$. Then $W(g^{u \cdot 2^j}) + V(g^{u \cdot 2^j})^x(g + r) = 0 = W(g^{u \cdot 2^k}) + V(g^{u \cdot 2^k})^x(g + r)$. It follows that

$$(g + r)^{2^{n-j}} = (W(g^u)/V(g^u)^x)^{2^n} = (g + r)^{2^{n-k}}.$$

Since $r \in K$, the above equality implies $g^{2^{n-j}} = g^{2^{n-k}}$. Since g is primitive, we have $(2^n - 1) | (2^{n-j} - 2^{n-k})$. It follows that $n | (k - j)$, which is a contradiction. □

The following lemma gives lower bound on $\epsilon(n, g, x, g)$ for any private key $x \in \mathbb{Z}_q$.

Lemma 3.2.4 *For every $x \in \mathbb{Z}_q$, we have that $\epsilon(n, g, x, g) \geq 1 - \dfrac{|U_x|}{|\mathbb{D}_{n,g,x}|}$.*

Proof: Due to Equality (3.2) and Equality (3.4), we have that

$$\epsilon(n,g,x,g) = \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{R\in\mathbb{D}_{n,g,x}}(1-\mathbf{H}_{x,s,r,g,R})}{2q\cdot|\mathbb{D}_{n,g,x}|} = \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{u\in U_x}\sum_{R\in\mathbf{D}_u}(1-\mathbf{H}_{x,s,r,g,R})}{2q\cdot|\mathbb{D}_{n,g,x}|}.$$

Let $s \in \mathbb{Z}_q$ and $r \in K$ be arbitrary. Due to Lemma 3.2.3, for every $u \in U_x$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$, or $E(t)$ has at most one root in $\mathbf{D}_u$. It follows that

$$\sum_{R\in\mathbf{D}_u}(1-\mathbf{H}_{x,s,r,g,R}) \geq |\mathbb{E}_u| - 1.$$

Therefore,

$$\epsilon(n,g,x,g) \geq \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{u\in U_x}(|\mathbb{E}_u|-1)}{2q\cdot|\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|}.$$

□

We shall bound $\epsilon(n,g,x,g)$ for various settings of the integer n and private key x. As the first case, we suppose that n is a prime and have the following lemma.

Lemma 3.2.5 *If n is prime, then $\epsilon(n,g,x,g) > 1 - \frac{2}{n}$ for every $x \in \mathbb{Z}_q$*

Proof: Due to Lemma 3.2.1, $|\mathbb{E}_u|$ is a divisor of n for every $x \in \mathbb{Z}_q$ and $u \in U_x$. Since n is prime, we have that $|\mathbb{E}_u| = 1$ or n.

- If $|U_x| = 1$, then $U_x = \{1\}$ due to Lemma 3.2.2. It is obvious that $|\mathbb{E}_1| = n$. By Lemma 3.2.4, we have

$$\epsilon(n,g,x,g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{1}{n} > 1 - \frac{2}{n}.$$

- If $|U_x| > 1$ and $0 \in U_x$, then we have that

$$\epsilon(n,g,x,g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{1+n(|U_x|-1)} > 1 - \frac{2}{n}.$$

- If $|U_x| > 1$ and $0 \notin U_x$, then we have that

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{n \cdot |U_x|} = 1 - \frac{1}{n} > 1 - \frac{2}{n}.$$

□

Below we lower bound $\epsilon(n, g, x, g)$ for **any** integer $n \geq 2$ and private key $x \in \mathbb{Z}_q$. For any positive integer $d|n$, we set

$$\lambda_{x.d} = |\{u : u \in U_x \text{ and } |\mathbb{E}_u| = d\}|.$$

Due to Lemma 3.2.2 and the requirements on database block R (imposed by Claim 3.2.1), $\lambda_x = (\lambda_{x,d})$ belongs to the following set

$$\Psi_n = \{z = (z_d)_{d|n} : 0 \leq z_1 \leq 1; 1 \leq z_n \leq N_2(n); 0 \leq z_d \leq N_2(d) \text{ for } d|n, 1 < d < n\},$$

where the coordinates of λ_x and z are indexed by positive divisors of n. Due to Lemma 3.2.4, we have that

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{\sum_{d|n} \lambda_{x,d}}{\sum_{d|n} d\lambda_{x,d}}. \tag{3.5}$$

We turn to upper bound the following function

$$\psi_n(z) = \frac{\sum_{d|n} z_d}{\sum_{d|n} dz_d},$$

on Ψ_n. Because this is relatively hard, we turn to upper bound the function

$$\phi_n(z) = \frac{\sum_{d=1}^{n} z_d}{\sum_{d=1}^{n} dz_d},$$

where $z = (z_1, \ldots, z_n)$ is taken from the following set

$$\Phi_n = \{z = (z_1, \ldots, z_n) : 0 \leq z_1 \leq 1; 1 \leq z_n \leq N_2(n); 0 \leq z_d \leq N_2(d) \text{ for } 1 < d < n\}.$$

Let $\omega(n)$ be the maximum value of $\phi_n(z)$ on Φ_n, i.e.,

$$\omega(n) = \max\{\phi_n(z) : z \in \Phi_n\}.$$

Lemma 3.2.6 *For every $x \in \mathbb{Z}_q$, we have that $\epsilon(n, g, x, g) \geq 1 - \omega(n)$.*

Proof: Clearly, $\omega(n) = \max\{\phi_n(z) : z \in \Phi_n\} \geq \max\{\psi_n(z) : z \in \Psi_n\} \geq \psi_n(\lambda_x)$. Due to Inequality (3.5), we have that $\epsilon(n, g, x, g) \geq 1 - \psi_n(\lambda_x) \geq 1 - \omega(n)$ for every $x \in \mathbb{Z}_q$. □

Due to Lemma 3.2.6, it is sufficient to upper bound $\omega(n)$.

Lemma 3.2.7 *Suppose that $\omega(n) = \phi_n(\xi)$ for $\xi = (\xi_1, \ldots, \xi_n) \in \Phi_n$. Then $\xi_1 = \xi_n = 1$. Furthermore, if $n \geq 3$, then there is an integer $1 < h < n$ such that $\xi_d = N_2(d)$ for every integer $1 < d \leq h$ and $\xi_d = 0$ for every integer $h < d < n$.*

Proof: It is trivial to verify that $\xi_1 = \xi_2 = 1$ for $n = 2$. From now on, we suppose that $n \geq 3$.

(a) For every $(0, z_2, \ldots, z_n), (1, z_2, \ldots, z_n) \in \Phi_n$, it is easy to see that

$$\phi_n(0, z_2, \ldots, z_n) - \phi_n(1, z_2, \ldots, z_n) < 0,$$

which implies that $\xi_1 = 1$.

(b) For every $(1, z_2, \ldots, z_{n-1}, z_n), (1, z_2, \ldots, z_{n-1}, 1) \in \Phi_n$ (where $z_n > 1$), it is easy to see that

$$\phi_n(1, z_2, \ldots, z_{n-1}, z_n) - \phi_n(1, z_2, \ldots, z_{n-1}, 1) < 0,$$

which implies that $\xi_n = 1$.

(c) Suppose $0 < \xi_h < N_2(h)$ for some integer $1 < h < n$. Let

$$C_1 = \sum_{d=1}^{h-1} \xi_d, C_2 = \sum_{d=h+1}^{n} \xi_d, C_3 = \sum_{d=1}^{h-1} d\xi_d, C_4 = \sum_{d=h+1}^{n} d\xi_d.$$

Then due to the maximality of $\omega(n)$, we have that

$$0 \geq \phi_n(\xi_1, \ldots, \xi_h + 1, \ldots, \xi_n) - \phi_n(\xi) = \frac{C_3 + C_4 - hC_1 - hC_2}{(C_3 + h(\xi_h + 1) + C_4)(C_3 + h\xi_h + C_4)};$$

$$0 \geq \phi_n(\xi_1, \ldots, \xi_h - 1, \ldots, \xi_n) - \phi_n(\xi) = \frac{-C_3 - C_4 + hC_1 + hC_2}{(C_3 + h(\xi_h - 1) + C_4)(C_3 + h\xi_h + C_4)}.$$

The above inequalities imply that $C_3 + C_4 = hC_1 + hC_2$. Hence, it follows that

$$h = \frac{\sum_{d=1}^{n} d\xi_d}{\sum_{d=1}^{n} \xi_d} = \frac{1}{\omega(n)}.$$

(d) We claim that $\xi_a = N_2(a)$ for every $1 < a < h$. Otherwise, by (c), we have that $\xi_a = 0$ and

$$\omega(n) < \phi_n(\xi_1, \ldots, \xi_a + 1, \ldots, \xi_h - 1, \ldots, \xi_n),$$

which is a contradiction.

(e) We claim that $\xi_b = 0$ for every $h < b < n$. Otherwise, by (c), we have that $\xi_b = N_2(b)$ and

$$\omega(n) < \phi_n(\xi_1, \ldots, \xi_h + 1, \ldots, \xi_b - 1, \ldots, \xi_n),$$

which is a contradiction.

(f) Finally, we show that $\omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1)$. Due to (c), (d) and (e), we have that

$$\xi = (1, N_2(2), \ldots, N_2(h-1), \xi_h, 0, \ldots, 0, 1).$$

Since $\phi_n(\xi) = \omega(n) \geq \phi_n(1, N_2(2), \ldots, N_2(h-1), 0, 0, \ldots, 0, 1)$, it follows that

$$hC_1 - C_3 \leq n - h.$$

If $hC_1 - C_3 < n - h$, then

$$\omega(n) < \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1),$$

which is a contradiction. Therefore, $hC_1 - C_3 = n - h$.Then it is not hard to verify that

$$\omega(n) = \phi_n(\xi) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1).$$

Therefore, we could have taken $\xi = (1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1)$.

□

Due to Lemma 3.2.7, for every integer $n \geq 3$, there is at least one integer $1 < h < n$ such that

$$\omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1). \tag{3.6}$$

Note that the integer h may be not unique. For every integer $n \geq 3$, we define

$$h(n) = \min \{h : \omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1), \text{ where } 1 < h < n\} \tag{3.7}$$

to be the smallest integer $1 < h < n$ such that Equality (3.6) holds. Next lemma shows that $h(n)$ is an increasing function of n.

Lemma 3.2.8 *We have that $h(n+1) \geq h(n)$ for every positive integer $n \geq 3$.*

Proof: Due to the definition of $h(n)$, i.e., Equality (3.7), it is not hard to see that

$$\phi_n(1, N_2(2), \ldots, N_2(l-1), N_2(l), 0, \ldots, 0, 1) > \phi_n(1, N_2(2), \ldots, N_2(l-1), 0, 0, \ldots, 0, 1)$$

for every integer $2 \leq l \leq h(n)$. Equivalently, we have that

$$\frac{1}{l} > \frac{\sum_{d=2}^{l-1} N_2(d) + 2}{\sum_{d=2}^{l-1} dN_2(d) + n + 1} \tag{3.8}$$

for every integer $2 \leq l \leq h(n)$. Due to Inequality (3.8), it is not hard to verify that

$$\phi_{n+1}(1, \ldots, N_2(l), 0, \ldots, 0, 1) > \phi_{n+1}(1, \ldots, 0, 0, \ldots, 0, 1) \tag{3.9}$$

for every integer $2 \leq l \leq h(n)$. In particular, Inequality (3.9) holds for $l = h(n)$. Hence, $h(n+1) \geq h(n)$. □

On the other hand, $\omega(n)$ is a decreasing function of n:

Lemma 3.2.9 *We have that $\omega(n+1) < \omega(n)$ for every positive integer $n \geq 3$.*

Proof: Due to Lemma 3.2.8, we have that $h(n+1) \geq h(n)$. If $h(n+1) = h(n)$, then

$$\omega(n+1) = \frac{\sum_{d=2}^{h(n+1)} N_2(d) + 2}{\sum_{d=2}^{h(n+1)} dN_2(d) + n + 2} = \frac{\sum_{d=2}^{h(n)} N_2(d) + 2}{\sum_{d=2}^{h(n)} dN_2(d) + n + 2} < \omega(n).$$

If $h(n+1) > h(n)$, then

$$\omega(n) = \frac{\sum_{d=2}^{h(n)} N_2(d) + 2}{\sum_{d=2}^{h(n)} dN_2(d) + n + 1} \geq \frac{1}{h(n)+1} \geq \omega(n+1),$$

where the second inequality and the third inequality follow from the definition of $h(n)$ and $h(n+1)$. □

We enumerate the values of $h(n)$ and $\omega(n)$ for some integers n in Table 3.2.

n	$h(n)$	$\omega(n)$	n	$h(n)$	$\omega(n)$	n	$h(n)$	$\omega(n)$
2	1	0.66667	12	4	0.24242	296	10	0.09996
3	1	0.50000	20	5	0.19718	522	11	0.09089
4	2	0.42857	34	6	0.16547	934	12	0.08332
5	2	0.37500	57	7	0.14236	1681	13	0.07692
6	2	0.33333	98	8	0.12478	3058	14	0.07143
7	3	0.31250	169	9	0.11101	5596	15	0.06667

Table 3.2: Values of $h(n)$ and $\omega(n)$

Lemma 3.2.10 *For every integer $n \geq 7$, we have that $\omega(n) \geq \dfrac{5}{n+9}$.*

Proof: Due to Table 3.2 and Lemma 3.2.8, we have that $h(n) \geq 3$ for every integer $n \geq 7$. It follows that $\omega(n) \geq \phi_n(1, 1, 2, 0, \ldots, 0, 1) = \dfrac{5}{n+9}$. □

Al last, we have the following theorem:

Theorem 3.2.1 *We have that*

$$\eta(n, g, g) \geq \begin{cases} 1 - \omega(n) & \text{if } 2 \leq n \leq 6 \text{ or } n \geq 7 \text{ is composite;} \\ 1 - \frac{2}{n} & \text{if } n \geq 7 \text{ is prime.} \end{cases}$$

Proof: Table 3.2 shows that $\omega(n) \leq 2/n$ for every integer $2 \leq n \leq 6$. Due to Lemma 3.2.5 and Lemma 3.2.6, we have that $\epsilon(n, g, x, g) \geq \max\{1-2/n, 1-\omega(n)\} = 1-\omega(n)$ for $n = 2, 3, 5$, and $\epsilon(n, g, x, g) \geq 1 - \omega(n)$ for $n = 4, 6$. Due to Equality (3.3), we have that

$$\eta(n, g, g) = \frac{1}{q} \sum_{x \in \mathbb{Z}_q} \epsilon(n, g, x, g) \geq 1 - \omega(n).$$

On the other hand, due to Lemma 3.2.5, Lemma 3.2.6 and Lemma 3.2.10, we have that $\epsilon(n, g, x, g) \geq \max\{1 - 2/n, 1 - \omega(n)\} = 1 - 2/n$ if $n \geq 7$ is prime and $\epsilon(n, g, x, g) \geq 1-\omega(n)$ if $n \geq 7$ is composite. Due to Equality (3.3), $\eta(n, g, g) \geq 1-2/n$ if $n \geq 7$ is prime and $\eta(n, g, g) \geq 1 - \omega(n)$ if $n \geq 7$ is composite. □

By Theorem 3.2.1, Lemma 3.2.9 and Table 3.2, we see that $\eta(n, g, g)$ is always non-negligible for every integer $n \geq 2$. Hence, we have the following theorem

Theorem 3.2.2 *The restricted version does not satisfy the correctness requirement whenever $F(t) = g$ is the polynomial $\mathcal{U}$ wants to evaluate.*

3.2.4 Extensions

Extension to a set of polynomials. We follow the notations in Section 3.2.3 and show that the restricted version does not satisfy the correctness requirement whenever $F(t) \in \mathcal{P}$, where

$$\mathcal{P} = \{f(t) = \sum_{k=0}^{d} f_k t^k : \exists\ 0 \leq l \leq d \text{ such that } f_l \in L \text{ is primitive and } f_k \in K \text{ for every } k \neq l\}.$$

Note that the polynomial $F(t) = g$ we studied in Section 3.2.3 is in $\mathcal{P}$ and satisfies Lemma 3.2.3, which is critical for obtaining all subsequent lemmas and theorems. Next lemma shows that Lemma 3.2.3 holds for any polynomial $F(t) \in \mathcal{P}$.

Lemma 3.2.11 *Let $F(t) \in \mathcal{P}$. Then for every $x \in \mathbb{Z}_q, u \in U_x, s \in \mathbb{Z}_q$ and $r \in K$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$ or $E(t)$ has at most one root in $\mathbf{D}_u$.*

Proof: If $V(g^u) = 0$, then $V(g^{u \cdot 2^j}) = V(g^u)^{2^j} = 0$ for every $j \in \mathbb{N}$, i.e., $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$. Otherwise, we have $V(\beta) \neq 0$ for every $\beta \in \mathbf{D}_u$. Suppose $F(t) = \sum_{k=0}^{d} F_k t^k$, where $F_l \in L$ is of order q and $F_k \in K$ for every $k \neq l$. We show that $E(t)$ has at most one root in $\mathbf{D}_u$, where $E(t) = W(t) + V(t)^x(F(t) + r)$. Suppose $E(t)$ has two different roots in $\mathbf{D}_u$, say $g^{u \cdot 2^a}$ and $g^{u \cdot 2^b}$ for $0 \leq a < b < n$. Then $W(g^{u \cdot 2^a}) + V(g^{u \cdot 2^a})^x(F(g^{u \cdot 2^a}) + r) = 0 = W(g^{u \cdot 2^b}) + V(g^{u \cdot 2^b})^x(F(g^{u \cdot 2^b}) + r)$. Hence,

$$(F(g^{u \cdot 2^a}) + r)^{2^{n-a}} = (F(g^{u \cdot 2^b}) + r)^{2^{n-b}}. \tag{3.10}$$

Let $c \in \{a, b\}$. Then it is not hard to see that

$$(F(g^{u \cdot 2^c}) + r)^{2^{n-c}} = \sum_{k=0}^{l-1} F_k g^{uk} + \sum_{k=l+1}^{d} F_k g^{uk} + F_l^{2^{n-c}} g^{ul} + r.$$

Due to Equality (3.10), we have that $F_l^{2^{n-a}} = F_l^{2^{n-b}}$. Since $F_l \in L$ is primitive, we have $(2^n - 1) | (2^{n-a} - 2^{n-b})$ and therefore $n | (b - a)$, which is a contradiction. $\square$

Due to Lemma 3.2.11, all lemmas and theorems subsequent to Lemma 3.2.3 in Section 3.2.3 can be generalized for any polynomial $F(t) \in \mathcal{P}$. Therefore, we have

Theorem 3.2.3 *The restricted version does not satisfy the correctness requirement whenever $F(t) \in \mathcal{P}$ is the polynomial $\mathcal{U}$ wants to evaluate.*

Clearly, the class $\mathcal{P}$ contains infinitely many polynomials over L. Specifically, the number of polynomials in $\mathcal{P}$ of degree $\leq d$ is equal to $d2^d\Phi(2^n - 1)$, where Φ is the

Euler totient function. Hence, the Bringer-Chabanne EPIR protocol fails frequently for a very large class of polynomials.

Extension to any characteristic. As stressed in Section 3.2.2, our methodology is applicable when the characteristic of all related finite fields is any prime p. For example, it is obvious that we have an analog of Lemma 3.2.6 when $p > 2$. Therefore, the following theorem holds as well.

Theorem 3.2.4 *We have that* $\eta(n, g, g) \geq 1 - \omega_p(n)$ *for every integer* $n \geq 2$*, where* $g \in \mathbb{F}_{p^n}$ *is primitive and* p *is an arbitrary prime number.* [1]

It follows that Theorem 3.2.3 also holds when the characteristic of all related finite fields is any prime $p > 2$.

[1] $\omega_p(n)$ is an analog of the function $\omega(n)$ when the characteristic of all related finite fields is p.

Chapter 4

Query-Efficient Locally Decodable Codes of Subexponential Length

In this chapter, we develop the algebraic theory behind the constructions of Yekhanin [96] and Efremenko [36], in an attempt to understand the "algebraic niceness". We show that Mersenne numbers that are products of two primes are algebraically nice. We also obtain new efficient LDCs and PIR protocols.

4.1 Efremenko's Framework

Let $m = p_1 \dots p_r$ be a product of $r \geq 2$ distinct odd primes $p_1, \dots, p_r$. Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$ and h be a positive integer. Let t be the multiplicative order of $2 \in \mathbb{Z}_m^*$, and $\gamma \in \mathbb{F}_{2^t}^*$ be a primitive mth root of unity. Efremenko's framework [36] for constructing LDCs is essentially a generalization of Yekhanin [96]. It consists of two basic ingredients: *S-matching family* and *S-decoding polynomial*:

Definition 4.1.1 (S-Matching Family) *For $S \subseteq \mathbb{Z}_m \setminus \{0\}$, a family of vectors $\{u_i\}_{i=1}^n \subseteq \mathbb{Z}_m^h$ is called an* S-matching family *if:*

- $\langle u_i, u_i \rangle_m = 0$ *for any integer* $i \in [n]$*;*
- $\langle u_i, u_j \rangle_m \in S$ *for any integers* $i, j \in [n]$ *such that* $i \neq j$*.*

Definition 4.1.2 (S-Decoding Polynomial) *For $S \subseteq \mathbb{Z}_m \backslash \{0\}$, a polynomial $P(X) \in \mathbb{F}_{2^t}[X]$ is called an* S-decoding polynomial *if*

- $P(\gamma^s) = 0$ *for any integer* $s \in S$;
- $P(1) = 1$.

The S-matching family and S-decoding polynomial yield linear LDCs immediately.

Theorem 4.1.1 ([36]) *Let $\{u_i\}_{i=1}^n \subseteq \mathbb{Z}_m^h$ be an S-matching family and $P(X) = a_0 + a_1 X^{b_1} + \ldots + a_{k-1} X^{b_{k-1}} \in \mathbb{F}_{2^t}[X]$ be an S-decoding polynomial with k monomials. Then there is a k-query linear* LDC $\mathbf{C} : \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}^{m^h}$ *depicted by Figure 4-1.*

Encoding
Let $e_j \in \mathbb{F}_{2^t}^n$ be the j-th unit vector for every $j \in [n]$. The coordinates of $\mathbf{C}(x)$ are indexed by vectors in $\mathbb{Z}_m^h$, where $x \in \mathbb{F}_{2^t}^n$. Below is the encoding algorithm.

- *for $j \in [n]$ and $v \in \mathbb{Z}_m^h$, set $\mathbf{C}(e_j)_v = \gamma^{\langle u_j, v \rangle_m}$;*
- *for $x = (x_1, \ldots, x_n) \in \mathbb{F}_{2^t}^n$, set $\mathbf{C}(x) = \sum_{j=1}^n x_j \cdot \mathbf{C}(e_j)$.*

Decoding
The S-decoding polynomial is $P(X)$. To recover x_i from a possibly corrupted codeword $y \in \mathbb{F}_{2^t}^{m^h}$ of a message x, the decoding algorithm proceeds as follows:

- *choose $v \leftarrow \mathbb{Z}_m^h$ and query the coordinates $y_v, y_{v+b_1 u_i}, \ldots, y_{v+b_{k-1} u_i}$;*
- *output $\gamma^{-\langle u_i, v \rangle_m} \cdot (a_0 \cdot y_v + a_1 \cdot y_{v+b_1 u_i} + \ldots + a_{k-1} \cdot y_{v+b_{k-1} u_i})$.*

Figure 4-1: *Efremenko's framework*

Theorem 4.1.1 shows that for any $S \subseteq \mathbb{Z}_m \setminus \{0\}$, an S-matching family of size n and an S-decoding polynomial with k monomials yield a k-query LDC which encodes messages of length n as codewords of length m^h. Given m and h, we prefer a large S-matching family and a sparse S-decoding polynomial in order to obtain efficient LDCs. Efremenko [36] suggested that S be taken as the *canonical set* of m:

Definition 4.1.3 (Canonical Set) *Let $m = p_1 \ldots p_r$ be the product of $r \geq 2$ distinct odd primes $p_1, \ldots, p_r$. The* canonical set *of m is defined to be $S = \left\{ s_\sigma \in \mathbb{Z}_m : s_\sigma \equiv \sigma_i \bmod p_i, \text{ for every} i \in [r], \text{ where } \sigma \in \{0,1\}^r \setminus \{\mathbf{0}\} \right\}$.*

The following proposition investigates the existence of S-matching families and S-decoding polynomials for any composite integer m.

Proposition 4.1.1 ([36]) *Let $m = p_1 \dots p_r$ be the product of $r \geq 2$ distinct odd primes $p_1, \dots, p_r$. Then the following statements hold.*

- *There is a positive constant $c = c(m)$, such that for every integer $h > 0$, there is an S-matching family in $\mathbb{Z}_m^h$ of size $n \geq \exp\left(c(\log h)^r/(\log\log h)^{r-1}\right)$.*
- *There is an S-decoding polynomial with at most 2^r monomials.*

Efremenko's linear LDCs of subexponential length follow from Theorem 4.1.1 and Proposition 4.1.1.

Theorem 4.1.2 ([36]) *For every integer $r \geq 2$, there is a linear $(k_r, \delta, k_r\delta)$-LDC of length $N_r = \exp(\exp(O(\sqrt[r]{\log n(\log\log n)^{r-1}})))$ for which $k_r \leq 2^r$. In particular, when $r = 2$, there is a linear $(3, \delta, 3\delta)$-LDC of length $N_2 = \exp(\exp(O(\sqrt{\log n \log\log n})))$.*

4.2 Composition Method

The query complexity of the LDCs given by Theorem 4.1.2 is only bounded by 2^r. The composition method [55] allows one to improve this upper bound. Let $m_1 = p_1 \dots p_r, m_2 = q_1 \dots q_l$ and $m = m_1 m_2$, where $r, l \geq 2$ and $p_1, \dots, p_r, q_1, \dots, q_l$ are distinct odd primes. Let t_1, t_2, and t be the multiplicative orders of 2 in $\mathbb{Z}_{m_1}^*$, $\mathbb{Z}_{m_2}^*$, and $\mathbb{Z}_m^*$, respectively. Let S_1, S_2, S be the canonical sets of m_1, m_2 and m, respectively. Due to Theorem 4.1.1 and Theorem 4.1.2, there are linear LDCs $\mathbf{C}_r$, $\mathbf{C}_l$ and $\mathbf{C}_{r+l}$ of query complexities $k_r \leq 2^r$, $k_l \leq 2^l$, and $k_{r+l} \leq 2^{r+l}$, respectively. Let $P_1(X)$ and $P_2(X)$ be the decoding polynomials for $\mathbf{C}_r$ and $\mathbf{C}_l$, respectively. Let γ_1, $\gamma_2, \gamma \in \mathbb{F}_{2^t}$ be of order m_1, m_2 and m, respectively. Clearly, there are integers μ and ν such that $\gamma_1 = \gamma^{\mu m_2}$ and $\gamma_2 = \gamma^{\nu m_1}$. Itoh et al. [55] observed that $P(X) = P_1(X^{\mu m_2})P_2(X^{\nu m_1})$ is an S-decoding polynomial for $\mathbf{C}_{r+l}$. This observation results in the following theorem.

Theorem 4.2.1 ([55] Composition Method) *There is a linear locally decodable code $\mathbf{C} : \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}^{N_{r+l}}$ of query complexity $k \leq k_r k_l$.*

Theorem 4.2.1 shows that if $k_r < 2^r$ or $k_l < 2^l$, then $\mathbf{C}_{r+l}$ essentially has a local decoding algorithm that makes less than 2^{r+l} queries. For every integer $r \geq 4$, applying Theorem 4.2.1 to Efremenko's 3-query LDC $\mathbf{C}_2$ (based on $m_1 = 511$) of length N_2 and k_{r-2}-query LDC $\mathbf{C}_{r-2}$ (based on $m_2 = q_1 \dots q_{r-2}$ such that $\gcd(m_1, m_2) = 1$) of length N_{r-2} gives:

Corollary 4.2.1 ([55]) *For every integer $r \geq 4$, there is a k-query linear locally decodable code of length N_r and query complexity $k \leq 3 \cdot 2^{r-2}$.*

Comparing with Theorem 4.1.2, Corollary 4.2.1 gives us a considerable saving of query complexity. The saving is due to Efremenko's 3-query linear LDC, which depends on an integer 511. An integer m is called *algebraically nice* if it is the product of r distinct odd primes and there is an S-decoding polynomial of less than 2^r monomials, where $r \geq 2$ and S is the canonical set of m. For every integer $r \geq 2$, we denote by $\mathbb{M}_r$ the set of algebraically nice integers that have r distinct prime factors. It is known that $511 \in \mathbb{M}_2$ [36] and $15 \notin \mathbb{M}_2$ [55]. It is an open problem [36, 55] to find new elements of $\mathbb{M}_2$.

4.3 Algebraically Nice Mersenne Numbers

In this section, we answer the open problem raised by [36, 55]. Let $m = pq$ for two distinct odd primes p and q. Let t be the order of 2 in $\mathbb{Z}_m^*$ and $\gamma \in \mathbb{F}_{2^t}^*$ be a primitive m-th root of unity. Let $S = \{1, s_{01}, s_{10}\}$ be the canonical set of m and

$$\mathcal{F} = \{f(X) \in \mathbb{F}_{2^t}[X] : f(\gamma) = f(\gamma^{s_{01}}) = f(\gamma^{s_{10}}) = 0 \text{ and } f(1) = 1\}.$$

be the set of all S-decoding polynomials. Clearly, the number k of monomials contained by any $f(X) \in \mathcal{F}$ can be as small as 4. However, k cannot be less than 3. Otherwise, we would have a 2-query LDC of subexponential length, which contradicts the well-known lower bound $\exp(\Omega(n))$ by Kerenidis et al. [59]. This fact is shown in the following proposition.

Proposition 4.3.1 *Each polynomial $f \in \mathcal{F}$ contains at least 3 monomials.*

Proof: Suppose that $f(X) = ax^u + bx^v \in \mathcal{F}$ contains less than three monomials. Then $a\gamma^u + b\gamma^v = a\gamma^{us_{01}} + b\gamma^{vs_{01}} = a\gamma^{us_{10}} + b\gamma^{vs_{10}} = 0$ and $a + b = 1$. It follows that $a\gamma^{u-v} = a\gamma^{(u-v)s_{01}} = a\gamma^{(u-v)s_{10}} = 1 + a$. Obviously, $a \neq 0$ and therefore $\gamma^{u-v} = \gamma^{(u-v)s_{01}} = \gamma^{(u-v)s_{10}}$. This implies $m|(u-v)$ and therefore $a = 1 + a$, which is a contradiction. □

Proposition 4.3.1 shows that the best we can expect is to have an S-decoding polynomial with exactly three monomials. Let

$$\mathcal{G} = \{g(X) \in \mathbb{F}_{2^t}[X] : g(\gamma) = g(\gamma^{s_{01}}) = g(\gamma^{s_{10}}) = 0 \text{ and } g(1) \neq 0\}.$$

Then it is easy to see that there is a polynomial $f(X) \in \mathcal{F}$ with three monomials if and only if there is a polynomial $g(X) \in \mathcal{G}$ with three monomials. We shall determine when $\mathcal{G}$ does contain a polynomial with 3 monomials in the remaining of this section. W.l.o.g, let $g(X) = X^u + aX^v + b \in \mathcal{G}$ have 3 monomials, where $u, v \in \mathbb{Z}_m \setminus \{0\}$ are distinct and $a, b \in \mathbb{F}_{2^t} \setminus \{0\}$. Clearly, we have that

$$\begin{pmatrix} \gamma^{us_{01}} & \gamma^{vs_{01}} & 1 \\ \gamma^{us_{10}} & \gamma^{vs_{10}} & 1 \\ \gamma^u & \gamma^v & 1 \end{pmatrix} \begin{pmatrix} 1 \\ a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{4.1}$$

$$1 + a + b \neq 0. \tag{4.2}$$

m	$M_{11} = 2^{11} - 1 = 2047$	$M_{23} = 2^{23} - 1 = 8388607$
$\mathbb{F}_{2^t}$	$\mathbb{F}_{2^{11}} = \mathbb{F}_2[\gamma]/(\gamma^{11} + \gamma^2 + 1)$	$\mathbb{F}_{2^{23}} = \mathbb{F}_2[\gamma]/(\gamma^{23} + \gamma^5 + 1)$
S	$\{1, 713, 1335\}$	$\{1, 5711393, 2677215\}$
f	$\gamma^{1485}X^{29} + \gamma^{694}X^{27} + \gamma^{118}$	$\gamma^{6526329}X^{3526} + \gamma^{7574532}X^{3363} + \gamma^{2861754}$

Table 4.1: Two new algebraically nice numbers

A computer search based on (4.1) and (4.2) shows that $M_{11} = 2047 = 23 \times 89$ and $M_{23} = 8388607 = 47 \times 178482$ are in $\mathbb{M}_2$. The S-decoding polynomials associated with these numbers are enumerated in Table 4.1.

Theorem 4.2.1 shows that the more numbers in $\mathbb{M}_2$ we find, the more improvements we get on the query complexity within Efremenko's framework. This motivates the consideration of the numbers taking the form of M_{11} and M_{23}, and to understand why they result in local decoding algorithms that make only 3 queries.

Theorem 4.3.1 *If $m = 2^t - 1 = pq$ for three primes t, p and q, then $m \in \mathbb{M}_2$.*

The proof of Theorem 4.3.1 is based on analysis of (4.1) and (4.2), and is an easy consequence of Proposition 4.3.2 and Proposition 4.3.3. Let

$$\mathcal{Z} = \left\{(z_1 + z_2)(z_1 z_2 + z_2)^{-1} : \ z_1, z_2 \in \mathbb{F}_{2^t}^*, \text{ord}(z_1) = p, \text{ and ord}(z_2) = q\right\}$$

be a multiset, where ord stands for the order of a field element.

Proposition 4.3.2 *If $\mathcal{Z}$ contains an element of multiplicity > 1, then $m \in \mathbb{M}_2$.*

Proof: Suppose $\mathcal{Z}$ contains an element of multiplicity > 1. Then there exist $z_1, z_2, z_1', z_2' \in \mathbb{F}_{2^t}^*$ such that $\text{ord}(z_1) = \text{ord}(z_1') = p, \text{ord}(z_2) = \text{ord}(z_2') = q, (z_1, z_2) \neq (z_1', z_2')$ and $(z_1 + z_2)(z_1 z_2 + z_2)^{-1} = (z_1' + z_2')(z_1' z_2' + z_2')^{-1}$. Because $\text{ord}(\gamma^{s_{10}}) = p$ and $\text{ord}(\gamma^{s_{01}}) = q$, there exist $u_1, v_1 \in \mathbb{Z}_p \setminus \{0\}$ and $u_2, v_2 \in \mathbb{Z}_q \setminus \{0\}$ such that $z_1 = \gamma^{u_1 s_{10}}, z_2 = \gamma^{u_2 s_{01}}, z_1' = \gamma^{v_1 s_{10}}$ and $z_2' = \gamma^{v_2 s_{01}}$. Due to the Chinese Remainder Theorem, there are integers $u, v \in \mathbb{Z}_m \setminus \{0\}$ such that $u \equiv u_1 \bmod p, u \equiv u_2 \bmod q; v \equiv v_1 \bmod p$ and $v \equiv v_2 \bmod q$. It follow that $z_1 = \gamma^{us_{10}}, z_2 = \gamma^{us_{01}}, z_1' = \gamma^{vs_{10}}, z_2' = \gamma^{vs_{01}}, u \neq v$ and

$$(\gamma^u + \gamma^{us_{01}})(\gamma^u + \gamma^{us_{10}})^{-1} = (\gamma^v + \gamma^{vs_{01}})(\gamma^v + \gamma^{vs_{10}})^{-1}. \tag{4.3}$$

The last condition implies that the matrix $\Gamma_{u,v}$ (defined below)

$$\begin{pmatrix} \gamma^{us_{01}} & \gamma^{vs_{01}} & 1 \\ \gamma^{us_{10}} & \gamma^{vs_{10}} & 1 \\ \gamma^{u} & \gamma^{v} & 1 \end{pmatrix}$$

is singular, i.e., $\mathrm{rank}(\Gamma_{u,v}) < 3$. On the other hand, it is clear that $\mathrm{rank}(\Gamma_{u,v}) \geq 1$. Hence, we have that $\mathrm{rank}(\Gamma_{u,v}) \in \{1, 2\}$. If $\mathrm{rank}(\Gamma_{u,v}) = 1$, then the rank of

$$\begin{pmatrix} \gamma^{us_{01}} + \gamma^{u} & \gamma^{vs_{01}} + \gamma^{v} & 0 \\ \gamma^{us_{10}} + \gamma^{u} & \gamma^{vs_{10}} + \gamma^{v} & 0 \\ \gamma^{u} & \gamma^{v} & 1 \end{pmatrix}$$

is also equal to 1. This implies that $\gamma^{us_{01}} = \gamma^{us_{10}} = \gamma^{u}$ and $\gamma^{vs_{01}} = \gamma^{vs_{10}} = \gamma^{v}$. It follows that $m | \gcd(u, v)$, which is a contradiction. Therefore, $\mathrm{rank}(\Gamma_{u,v}) = 2$. As a consequence, (4.1) has a unique solution $(a, b) \in \mathbb{F}_{2^t}^2$.

We claim that $ab \neq 0$. If $a = 0$, then $b = \gamma^{us_{01}} = \gamma^{us_{10}} = \gamma^{u}$ due to (4.1), which implies that $u \equiv 0 \bmod m$, a contradiction. If $b = 0$, then $a = \gamma^{(u-v)s_{01}} = \gamma^{(u-v)s_{10}} = \gamma^{u-v}$ due to (4.1), which implies that $u \equiv v \bmod m$, also a contradiction.

Let $g(X) = X^u + aX^v + b$. So far, we learn that $g(X)$ has exactly three monomials and $g(\gamma) = g(\gamma^{s_{01}}) = g(\gamma^{s_{10}}) = 0$. We claim that $g(1) \neq 0$. Otherwise, $(1, 1, 1)$ is in the row space of $\Gamma_{u,v}$ and therefore the matrix

$$\begin{pmatrix} \gamma^{us_{01}} & \gamma^{vs_{01}} & 1 \\ \gamma^{us_{10}} & \gamma^{vs_{10}} & 1 \\ \gamma^{u} & \gamma^{v} & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

has rank two as well. Applying elementary row operations (adding the third row to each of the first three rows), we have that

$$(1 + \gamma^{u})(1 + \gamma^{v})^{-1} = (1 + \gamma^{us_{10}})(1 + \gamma^{vs_{10}})^{-1} = (1 + \gamma^{us_{01}})(1 + \gamma^{vs_{01}})^{-1}. \qquad (4.4)$$

Due to (4.3) and (4.4), we have that $\gamma^{(u-v)s_{01}} = \gamma^{(u-v)s_{10}}$, which implies $u = v$, a contradiction.

We have shown $g(X) \in \mathcal{G}$ and has exactly three monomials. Hence, there is a polynomial $f(X) \in \mathcal{F}$ with three monomials. We conclude that $m \in \mathbb{M}_2$. $\square$

Proposition 4.3.3 *If $m = 2^t - 1 = pq$ for three primes t, p and q, then $\mathcal{Z}$ contains an element of multiplicity > 1.*

Proof: Suppose that each element of $\mathcal{Z}$ has multiplicity one. Then $\mathcal{Z}$ is a set of cardinality $(p-1)(q-1)$. Let $G = \mathbb{F}_{2^t}^*$. Because $(z_1 + z_2)(z_1 z_2 + z_2)^{-1} = 1 + (1 + z_2^{-1})(1 + z_1^{-1})^{-1}$ for $z_1, z_2 \in G$, we have that

$$T = \left\{(1 + z_2)(1 + z_1)^{-1} : z_1, z_2 \in G, \mathrm{ord}(z_1) = p, \mathrm{ord}(z_2) = q\right\} \tag{4.5}$$

is also a set of cardinality $(p-1)(q-1)$. Let the following subsets

$$\begin{aligned} A &= \{1 + z_1 : z_1 \in G, \mathrm{ord}(z_1) = p\}, \\ B &= \{1 + z_2 : z_2 \in G, \mathrm{ord}(z_2) = q\}, \end{aligned} \tag{4.6}$$

of G be identified with two elements of the group ring $\mathbb{Z}[G]$.

We claim that $T \cup A^{(-1)} \cup B \cup \{1\} = G$. Because $|T| + |A^{-1}| + |B| + 1 = |G|$, it is sufficient to show that T, $A^{(-1)}$, B, and $\{1\}$ are pairwise disjoint. Clearly, $1 \notin T \cup A^{(-1)} \cup B$. If $T \cap A^{(-1)} \neq \emptyset$, then there exist $z_1, z_1', z_2 \in G$ such that $(1 + z_2)(1 + z_1)^{-1} = (1 + z_1')^{-1}$, where $\mathrm{ord}(z_1) = \mathrm{ord}(z_1') = p$ and $\mathrm{ord}(z_2) = q$. It follows that $(1 + z_2^2)(1 + z_1)^{-1} = (1 + z_2)(1 + z_1')^{-1}$. This implies that T has an element of multiplicity > 1, which is a contradiction. Similarly, we have that $T \cap B = A^{(-1)} \cap B = \emptyset$. Due to the claimed equality, we have

$$(A + 1)^{(-1)}(B + 1) = G. \tag{4.7}$$

Let $\gamma_p, \gamma_q \in G$ be of order p and q, respectively. Then $G = \{\gamma_p^\mu \gamma_q^\nu : \mu \in \mathbb{Z}_p, \nu \in \mathbb{Z}_q\}$. Let $\chi_p \in \widehat{G}$ be of order p such that $\chi_p(\gamma_p) = \theta_p$, where $\theta_p \in \mathbb{C}^*$ is of order p. Then

for every $\mu \in \mathbb{Z}_p$ and $\nu \in \mathbb{Z}_q$, $\chi_p(\gamma_p^\mu \gamma_q^\nu) = \theta_p^\mu \equiv 1 \bmod (1-\theta_p)$. It follows that $\chi_p(B+1) \equiv q \bmod (1-\theta_p)$. We claim that there exist a permutation $a : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ and a mapping $b : \mathbb{Z}_p^* \to \mathbb{Z}_q$ such that for every $i \in \mathbb{Z}_p^*$,

$$1 + \gamma_p^i = \gamma_p^{a(i)} \gamma_q^{b(i)}. \tag{4.8}$$

Applying χ_p to (4.7) gives $\chi_p(A+1)\chi_p(B+1) = 0$. If $\chi_p(B+1) = 0$, then $q \equiv \chi_p(B+1) \equiv 0 \bmod (1-\theta_p)$. On the other hand, $p = \sum_{i=1}^{p-1}(\theta_p^i - 1) \equiv 0 \bmod (1-\theta_p)$. Since $\gcd(p,q) = 1$, there are rational integers λ, ρ such that $\lambda p + \rho q = 1$. It follows that $1 \in (1-\theta_p)\mathbb{Z}[\theta_p]$, which is a contradiction because $(1-\theta_p)\mathbb{Z}[\theta_p]$ is a prime ideal in $\mathbb{Z}[\theta_p]$ (cf. [88, Lemma 1.4]). Hence, we have that $\chi_p(A+1) = 0$. Let $a : \mathbb{Z}_p^* \to \mathbb{Z}_p$ and $b : \mathbb{Z}_p^* \to \mathbb{Z}_q$ be two mappings such that $1 + \gamma_p^i = \gamma_p^{a(i)} \gamma_q^{b(i)}$ for every $i \in \mathbb{Z}_p^*$. It follows that $0 = \chi_p(A+1) = \sum_{i=1}^{p-1} \theta_p^{a(i)} + 1$. Since any $p-1$ elements of $\{1, \theta_p, \ldots, \theta_p^{p-1}\}$ form an integral basis of $\mathbb{Z}[\theta_p]$ over $\mathbb{Z}$, a must be a permutation of $\mathbb{Z}_p^*$. The expected equation (4.8) follows. Similarly, there exist a permutation $c : \mathbb{Z}_q^* \to \mathbb{Z}_q^*$ and a mapping $d : \mathbb{Z}_q^* \to \mathbb{Z}_p$ such that, for every $j \in \mathbb{Z}_q^*$,

$$1 + \gamma_q^j = \gamma_q^{c(j)} \gamma_p^{d(j)}. \tag{4.9}$$

Let $\chi_m \in \widehat{G}$ be of order m such that $\chi_m(\gamma_p) = \theta_p$ and $\chi_m(\gamma_q) = \theta_q$, where $\theta_p, \theta_q \in \mathbb{C}^*$ are of order p and q respectively. Applying χ_m to (4.7) gives $\chi_m(A+1)\chi_m(B+1) = 0$. If $\chi_m(A+1) = 0$, then $0 = \sum_{i=1}^{p-1} \chi_m(1+\gamma_p^i) + 1 = \sum_{i=1}^{p-1} \theta_p^{a(i)} \theta_q^{b(i)} + 1 = \sum_{i=1}^{p-1} \theta_p^{a(i)}(\theta_q^{b(i)} - 1)$. Since $\{\theta_p, \ldots, \theta_p^{p-1}\}$ is an integral basis of $\mathbb{Z}[\theta_p, \theta_q]$ over $\mathbb{Z}[\theta_q]$, we have $\theta_q^{b(i)} - 1 = 0$ for every $i \in \mathbb{Z}_p^*$. It follows that $1 + \gamma_p^i = \gamma_p^{a(i)}$ for every $i \in \mathbb{Z}_p^*$. Hence, $\{0, 1, \gamma_p, \ldots, \gamma_p^{p-1}\}$ is a subfield of $\mathbb{F}_{2^t}$. Hence, either $p+1 = 2$ or $p+1 = 2^t$, which is a contradiction. Similarly, if $\chi_m(B+1) = 0$, then we have that $\{0, 1, \gamma_q, \ldots, \gamma_q^{q-1}\}$ is a subfield of $\mathbb{F}_{2^t}$, which results in the same contradiction.

Hence, our original assumption about $\mathcal{Z}$ is wrong and the proposition is true. □

Proof: (Proof of Theorem 4.3.1) It is easy to see that the odd primes p and q are distinct. Theorem 4.3.1 follows from Proposition 4.3.2 and Proposition 4.3.3. □

Let $\mathbb{M}_{2,\text{Mersenne}} = \{m : m = 2^t - 1 = pq, \text{ where } t, p \text{ and } q \text{ are primes}\}$. Theorem 4.3.1 shows that $\mathbb{M}_{2,\text{Mersenne}}$ is a subset of $\mathbb{M}_2$ and therefore gives us a new family of algebraically nice integers.

m	p	m	p
M_{11}	23	M_{727}	176062917118154340379348818723316116770777491166445300472749449436575622328171096762726546652185892
M_{23}	47		
M_{37}	223		
M_{41}	13367		
M_{59}	179951	M_{809}	41483867312606056475251865474888423964616257742413275679781 37
M_{67}	193707721		
M_{83}	167		
M_{97}	11447	M_{881}	26431
M_{101}	7432339208719	M_{971}	2391710497317390956691632101601188504196248632150251 3
M_{103}	2550183799		
M_{109}	745988807		
M_{131}	263	M_{983}	1808226257914551209964473260866417929207023
M_{137}	32032215596496435569		
M_{139}	5625767248687	M_{997}	1675608165140848194887377679762631504050951915547329 02607
M_{149}	86656268566282183151		
M_{167}	2349023		
M_{197}	7487	M_{1063}	1485761479
M_{199}	164504919713	M_{1427}	1905458056472554697419312683097859050 3
M_{227}	26986333437777017		
M_{241}	22000409	M_{1487}	24464753918382797416777
M_{269}	13822297	M_{1637}	81679753
M_{271}	15242475217	M_{2927}	1217183584262023230020873
M_{281}	80929	M_{3079}	2532484664981064888738318 0721
M_{293}	401223624556162219711223 53		
M_{347}	14143189112952632419639	M_{3259}	2192680587227006249681922 1124452121
M_{373}	25569151		
M_{379}	180818808679	M_{3359}	6719
M_{421}	614002928307599	M_{4243}	101833
M_{457}	150327409	M_{4729}	6194418998141586667111247947 7273
M_{487}	4871		
M_{523}	160188778313202118610543685368878688932828701136501444932217468039063	M_{5689}	919724609777
		M_{6043}	1115552064241903805636990318 3
		M_{7331}	458072843161

Table 4.2: Fifty algebraically nice Mersenne numbers

We identify 50 numbers in $\mathbb{M}_{2,\text{Mersenne}}$ by computer search with the largest one being $M_{7331} = 2^{7331} - 1$. These numbers and their smallest prime divisors are enumerated in Table 4.2, where each number is of the form $m = M_t = 2^t - 1 = pq$. As an immediate corollary of Table 4.2, we have that

Proposition 4.3.4 $|\mathbb{M}_{2,\text{Mersenne}}| \geq 50$.

Proposition 4.3.5 *The numbers in* $\mathbb{M}_{2,\text{Mersenne}} \bigcup \{511\}$ *are pairwise coprime.*

Proof: Let $m = 2^t - 1 = pq \in \mathbb{M}_{2,\text{Mersenne}}$ be arbitrary. It is easy to see that t is the order of 2 in $\mathbb{Z}_p^*, \mathbb{Z}_q^*$ and $\mathbb{Z}_m^*$.

If $M_{t_1}, M_{t_2} \in \mathbb{M}_{2,\text{Mersenne}}$ are not coprime, then they have a common prime factor, say p. It follows that both t_1 and t_2 are equal to the order of 2 in $\mathbb{Z}_p^*$, i.e., we have $M_{t_1} = M_{t_2}$. If $M_t = 2^t - 1 \in \mathbb{M}_{2,\text{Mersenne}}$ and 511 are not coprime, then $7|M_t$ or $73|M_t$. It follows that $3|t$ or $9|t$. However, both cases are impossible since t is prime and M_t has two prime factors. □

Corollary 4.3.1 *There are at least 51 numbers in* $\mathbb{M}_2$ *that are pairwise coprime.*

The first 33 numbers in $\mathbb{M}_{2,\text{Mersenne}}$ are $M_{11}, M_{23}, \ldots, M_{809}$. It is not known whether M_{881} is the 34th number in $\mathbb{M}_{2,\text{Mersenne}}$. It is an interesting open problem to determine how many numbers $\mathbb{M}_{2,\text{Mersenne}}$ contains. A similar but much more well-known problem in number theory is determining the number of Mersenne primes. It is generally believed that there are infinitely many Mersenne primes. Compared with Mersenne primes, it seems reasonable to conjecture that $|\mathbb{M}_{2,\text{Mersenne}}| = \infty$.

Itoh et al. [55] showed that $15 \notin \mathbb{M}_2$ by exhaustive search. The following proposition allows us to give a computer-free proof of the same result.

Proposition 4.3.6 *The Mersenne number* $m = 2^t - 1 = pq$ *is in* $\mathbb{M}_2$ *if and only if there are coset representatives* α, β *and integers* $0 \leq c, d < t$ *such that*

(1) *The cyclotomic cosets* $\mathbb{E}_\alpha, \mathbb{E}_\beta$ *of* 2 *modulo* m *do not contain multiples of* p *or* q*;*

(2) $(\alpha, c) \neq (\beta, d)$*;*

(3) $\left(\frac{\gamma^{\alpha}+\gamma^{\alpha s_{01}}}{\gamma^{\alpha}+\gamma^{\alpha s_{10}}}\right)^{2^c} = \left(\frac{\gamma^{\beta}+\gamma^{\beta s_{01}}}{\gamma^{\beta}+\gamma^{\beta s_{10}}}\right)^{2^d}$.

Proof: Suppose $m \in \mathbb{M}_2$. Then there is a polynomial $g(X) = X^u + aX^v + b \in \mathcal{G}$ with exactly three monomials, where $u, v \in \mathbb{Z}_m \setminus \{0\}$ are distinct and $a, b \in \mathbb{F}_{2^t} \setminus \{0\}$. Clearly, (4.1) and (4.2) are satisfied. It follows that $\det(\Gamma_{u,v}) = 0$ and therefore

$$(\gamma^u + \gamma^{us_{01}})(\gamma^v + \gamma^{vs_{10}}) = (\gamma^u + \gamma^{us_{10}})(\gamma^v + \gamma^{vs_{01}}). \tag{4.10}$$

W.l.o.g, suppose that $u \in \mathbb{E}_{\alpha}$ and $v \in \mathbb{E}_{\beta}$, where $\mathbb{E}_{\alpha}$ and $\mathbb{E}_{\beta}$ are cyclotomic cosets of 2 modulo m. Suppose that $hp \in \mathbb{E}_{\alpha}$ for an integer h. Then $u \equiv 2^l hp \bmod m$ for an integer l since $u \in \mathbb{E}_{\alpha}$. Clearly, we have that $q \nmid h$ and $\gamma^u + \gamma^{us_{01}} = (\gamma^{hp} + \gamma^{hps_{01}})^{2^l} = 0$. Due to (4.10), we have $(\gamma^u + \gamma^{us_{10}})(\gamma^v + \gamma^{vs_{01}}) = 0$. Since $\gamma^u + \gamma^{us_{10}} = (\gamma^{hp} + \gamma^{hps_{10}})^{2^l} \neq 0$, we have that $\gamma^v + \gamma^{vs_{01}} = 0$, which implies $p|v$. Thus, $\gamma^{us_{10}} = \gamma^{2^l hps_{10}} = (\gamma^{hps_{10}})^{2^l} = 1$ and $\gamma^{vs_{10}} = (\gamma^{ps_{10}})^{v/p} = 1$, i.e., the second row of $\Gamma_{u,v}$ is $(1, 1, 1)$. We have that $1 + a + b = 0$ due to (4.1) and $1 + a + b \neq 0$ due to (4.2), which is a contradiction. Hence, $\mathbb{E}_{\alpha}$ does not contain any multiples of p. Using similar arguments, we can prove that (1) is true. To avoid unnecessary repetitions, we omit the details. Let the integers $0 \leq c, d < t$ be such that $u \equiv 2^c \alpha \bmod m$ and $v \equiv 2^d \beta \bmod m$. Due to the fact $u \neq v$ and (4.10), it is trivial to see that (2) and (3) hold.

Conversely, suppose that (1), (2) and (3) hold. Let $u \equiv 2^c \alpha \bmod m, v \equiv 2^d \beta \bmod m$, $z_1 = \gamma^{us_{10}}$, $z_2 = \gamma^{us_{01}}$, $z_1' = \gamma^{vs_{10}}$, and $z_2' = \gamma^{vs_{01}}$. Then it is easy to verify that $u, v \in \mathbb{Z}_m \setminus \{0\}$ are distinct, $\mathrm{ord}(z_1) = \mathrm{ord}(z_1') = p$, $\mathrm{ord}(z_2) = \mathrm{ord}(z_2') = q$ and $(z_1, z_2) \neq (z_1', z_2')$. (3) implies that

$$(z_1 + z_2)(z_1 z_2 + z_2)^{-1} = (z_1' + z_2')(z_1' z_2' + z_2')^{-1}, \tag{4.11}$$

i.e., $\mathcal{Z}$ contains an element of multiplicity > 1. Due to Proposition 4.3.2, we have that $m \in \mathbb{M}_2$. □

Corollary 4.3.2 *The integer 15 is not algebraically nice, i.e.* $15 \notin \mathbb{M}_2$.

Proof: Clearly, 2 is of order 4 in $\mathbb{Z}_{15}^*$ and the canonical set of 15 is $S = \{1, 6, 10\}$. Let $\mathbb{F}_{2^4} = \mathbb{F}_2[\gamma]/(\gamma^4 + \gamma + 1)$ and $\gamma \in \mathbb{F}_{2^4}^*$ be of order 15. The cyclotomic cosets of 2 modulo 15 are $\mathbb{E}_0 = \{0\}$, $\mathbb{E}_1 = \{1, 2, 4, 8\}$, $\mathbb{E}_3 = \{3, 6, 9, 12\}$, $\mathbb{E}_5 = \{5, 10\}$, and $\mathbb{E}_7 = \{7, 14, 13, 11\}$. Suppose $15 \in \mathbb{M}_2$. Then there exist coset representatives $\alpha, \beta \in \{1, 7\}$ and integers $0 \leq c, d < 4$ such that the items (1), (2) and (3) in Proposition 4.3.6 hold. If $\{\alpha, \beta\} = \{1\}$, then $\gamma^{3 \cdot 2^c} = ((\gamma + \gamma^6)/(\gamma + \gamma^{10}))^{2^c} = ((\gamma + \gamma^6)/(\gamma + \gamma^{10}))^{2^d} = \gamma^{3 \cdot 2^d}$ due to (3). It follows that $c = d$ and therefore $(\alpha, c) = (\beta, d)$, which contradicts (2). For $\{\alpha, \beta\} = \{7\}$ or $\{1, 7\}$, we have similar arguments and omit the details. □

4.4 Improved LDCs and PIR Protocols

In this section, we give new families of query-efficient LDCs and efficient PIR protocols using the algebraically nice Mersenne numbers in $\mathbb{M}_{2,\text{Mersenne}}$. Compared with [36] and [55], our LDCs and PIR protocols achieve considerable improvements of efficiency.

Query-Efficient LDCs. Let $N_r = \exp(\exp(O(\sqrt[r]{\log n (\log\log n)^{r-1}})))$ for any integers $r \geq 2$ and $n > 0$. By Corollary 4.3.1, Theorem 4.1.2, Theorem 4.2.1 and Table 4.1, we have

Theorem 4.4.1 *The following statements hold*

(1) *For every positive integer $r \leq 103$, there is a linear* LDC *of length N_r and query complexity*

$$k \leq \begin{cases} (\sqrt{3})^r, & \text{if } r \text{ is even} \\ 8 \cdot (\sqrt{3})^{r-3}, & \text{if } r \text{ is odd.} \end{cases}$$

(2) *For every integer $r \geq 104$, there is a linear* LDC *of length N_r and query complexity $k \leq (3/4)^{51} \cdot 2^r$.*

(3) *If $|\mathbb{M}_{2,\text{Mersenne}}| = \infty$, then for every integer $r \geq 1$, there is a linear* LDC *of length N_r and the same query complexity as in* (1).

Proof: (1) Let $r \in [103]$ be even. By Corollary 4.3.1, we can take pairwise coprime integers $m_1, \ldots, m_{r/2} \in \mathbb{M}_2$. Then there is a 3-query linear LDC of length N_2 based on each of them due to Theorem 4.1.2. Applying Theorem 4.2.1 $r/2 - 1$ times, we obtain a linear LDC of length N_r and query complexity $k \leq 3^{r/2}$.

Let $r \in [103]$ be odd. If $r = 1$, then the Hadamard code is a 2-query linear LDC of length $N_1 = \exp(n)$. If $r \geq 3$, then $r = 2 \cdot \frac{r-3}{2} + 3$. We can take $(r-1)/2$ pairwise coprime integers $m_1, \ldots, m_{\frac{r-1}{2}}$, where $m_1, \ldots, m_{\frac{r-3}{2}} \in \mathbb{M}_2$ and $m_{\frac{r-1}{2}}$ is a product of three distinct odd primes. By Theorem 4.1.2, there are a 3-query linear LDC of length N_2 based on each of $m_1, \ldots, m_{\frac{r-3}{2}}$ and a linear LDC of length N_3 and query complexity $k_3 \leq 2^3$ based on $m_{\frac{r-1}{2}}$. Applying Theorem 4.2.1 $(r-3)/2$ times gives a linear LDC of length N_r and query complexity $k \leq 3^{\frac{r-3}{2}} \cdot 8$.

(2) If $r \geq 104$, we can take 52 pairwise coprime integers $m_1, \ldots, m_{52}$, where $m_1, \ldots, m_{51} \in \mathbb{M}_2$ and m_{52} a product of $r - 102$ distinct odd primes. By Theorem 4.1.2, there is a 3-query linear LDC of length N_2 based on each of $m_1, \ldots, m_{51}$ and a linear LDC of length N_{r-102} and query complexity $\leq 2^{r-102}$ based on m_{52}. Applying Theorem 4.2.1 on these codes gives a linear LDC of length N_r and query complexity $k \leq 3^{51} \cdot 2^{r-102}$.

(3) It suffices to prove for $r \geq 104$. If r is even, we take $r/2$ pairwise coprime integers from $\mathbb{M}_{2,\text{Mersenne}}$. Otherwise, we take $(r-1)/2$ pairwise coprime integers where the first $(r-3)/2$ of them belong to $\mathbb{M}_{2,\text{Mersenne}}$ and the last one is a product of three distinct odd primes. Applying Theorem 4.2.1 to the LDCs based on these integers gives us the expected result. □

Efficient PIR Protocols. The new query-efficient LDCs can be turned into PIR protocols that are superior to [36] and [55]. Katz et al. [57] were the first to show generic transformations between information-theoretic PIR protocols and LDCs. Trevisan [87] introduced the notion of perfectly smooth decoders:

Definition 4.4.1 ([87]) *A k-query* LDC $\mathbf{C} : \Sigma^n \to \Gamma^N$ *is said to have a* perfectly smooth decoder *if it has a local decoding algorithm* $\mathbf{D}$ *satisfying:*

- *In every invocation, each query of* $\mathbf{D}$ *is uniformly distributed over* $[N]$.

- *For every* $x \in \Sigma^n$ *and* $i \in [n]$, $\Pr[\mathbf{D}^{\mathbf{C}(x)}(i) = x_i] = 1$.

LDCs with perfectly smooth decoders yield information-theoretic PIR protocols.

Proposition 4.4.1 ([87]) *If there is a* k*-query* LDC $\mathbf{C} : \Sigma^n \to \Gamma^N$ *with perfectly smooth decoder, then there is a* k*-server PIR protocol of communication complexity* $k(\log N + \log |\Gamma|)$.

The LDCs obtained by [36] and [55] both have perfectly smooth decoders, and so do our query-efficient LDCs. Applying Proposition 4.4.1 to the LDCs of [55], one obtains a family of positive integers $\{k^{(r)}\}_{r \geq 4}$, where $k^{(r)} \leq 3 \cdot 2^{r-2}$ for every integer $r \geq 4$. [55] shows that for each $r \geq 4$, there is a $k^{(r)}$-server PIR protocol whose communication complexity is $\exp(O(\sqrt[s]{\log n (\log \log n)^{s-1}}))$, where $s = \log k^{(r)} + 2 - \log 3$. Due to Theorem 4.4.1 and Proposition 4.4.1, we have the following improvements on their PIR protocols.

Theorem 4.4.2 *The following statements hold:*

- *There is a family of positive integers* $\{k^{\langle r \rangle}\}_{1 \leq r \leq 103}$ *for which* $k^{\langle r \rangle} \leq (\sqrt{3})^r$ *if* r *is even, and* $k^{\langle r \rangle} \leq 8 \cdot (\sqrt{3})^{r-3}$ *if* r *is odd, such that for every* $r \in [103]$*, there is a* $k^{\langle r \rangle}$*-server* PIR *of communication complexity* $\exp(O(\sqrt[s]{\log n (\log \log n)^{s-1}}))$*, where* $s = 2 \log k^{\langle r \rangle} / \log 3$ *if* r *is even, and* $s = (2 \log k^{\langle r \rangle} - 6 + 3 \log 3) / \log 3$ *if* r *is odd.*

- *There is a family of positive integers* $\{k^{\langle r \rangle}\}_{r \geq 104}$ *for which* $k^{\langle r \rangle} \leq (3/4)^{51} \cdot 2^r$*, such that for every* $r \geq 104$ *there is a* $k^{\langle r \rangle}$*-server* PIR *of communication complexity* $\exp(O(\sqrt[s]{\log n (\log \log n)^{s-1}}))$*, where* $s = \log k^{\langle r \rangle} + 102 - 51 \log 3$.

- *If* $|\mathbb{M}_{2,\text{Mersenne}}| = \infty$*, then there is a family of positive integers* $\{k^{\langle r \rangle}\}_{r \geq 1}$ *for which* $k^{\langle r \rangle} \leq (\sqrt{3})^r$ *if* r *is even, and* $k^{\langle r \rangle} \leq 8 \cdot (\sqrt{3})^{r-3}$ *if* r *is odd, such that for every* $r \geq 1$ *there is a* $k^{\langle r \rangle}$*-server* PIR *of communication complexity* $\exp(O(\sqrt[s]{\log n (\log \log n)^{s-1}}))$*, where* $s = 2 \log k^{\langle r \rangle} / \log 3$ *if* r *is even, and* $s = (2 \log k^{\langle r \rangle} - 6 + 3 \log 3) / \log 3$ *if* r *is odd.*

Chapter 5

Communication-Efficient Distributed Oblivious Transfer

In this chapter, we construct DOT protocols of sublinear communication complexity between the receiver and servers. We obtain both a specific reduction and a general reduction from DOT to information-theoretic PIR.

5.1 Information Equalities

Recall our definition of (t, τ)-private (k, l)-DOT-$\binom{n}{1}$ in Section 2.2.4 and the related random variables and assumptions over there. In this section, we shall prove several information equalities on the random variables. These equalities capture the essence of our definition of privacy and allow us to simplify the proofs.

The first of these equalities shows that a set of corrupted servers can learn information on what the receiver is interested in only from the queries they may obtain from the receiver. Intuitively, this is true because the messages the corrupted servers may obtain are from either the sender or the receiver. However, the messages from the sender are independent of those from the receiver.

Lemma 5.1.1 *In a one-round* (k, l)-DOT-$\binom{n}{1}$ *protocol, we have that* $H(\mathbf{I}|\mathbf{D}_T, \mathbf{Q}_T) = H(\mathbf{I}|\mathbf{Q}_T)$ *for every* $T \subseteq [l]$.

Proof: It is equivalent to show that $H(\mathbf{D}_T|\ \mathbf{Q}_T) = H(\mathbf{D}_T|\mathbf{Q}_T, \mathbf{I})$. Due to (2.5), we have $H(\mathbf{Q}_T|\mathbf{I}, \mathbf{Y}) = 0$ and therefore $H(\mathbf{D}_T) \geq H(\mathbf{D}_T|\mathbf{Q}_T) \geq H(\mathbf{D}_T|\mathbf{Q}_T, \mathbf{I}) \geq H(\mathbf{D}_T|\mathbf{Y}, \mathbf{I})$. Hence, it suffices to show $H(\mathbf{D}_T) = H(\mathbf{D}_T|\mathbf{Y}, \mathbf{I})$. Due to (2.2), we have that $H(\mathbf{D}_T|\mathbf{Y}) - H(\mathbf{D}_T|\mathbf{Y},\ \mathbf{I}) = H(\mathbf{I}|\mathbf{Y}) - H(\mathbf{I}|\mathbf{Y}, \mathbf{D}_T) = 0$ and therefore $H(\mathbf{D}_T|\mathbf{Y}, \mathbf{I}) = H(\mathbf{D}_T|\mathbf{Y})$. We turn to show that $H(\ \mathbf{D}_T) = H(\mathbf{D}_T|\mathbf{Y})$. Due to (2.4) and (2.1), we have that $H(\mathbf{Y}) \geq H(\mathbf{Y}|\ \mathbf{D}_T) \geq H(\mathbf{Y}|\mathbf{W}, \mathbf{X}) = H(\mathbf{Y})$, which implies $H(\mathbf{D}_T) - H(\mathbf{D}_T|\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{D}_T) = 0$. □

The second equality shows that before the transfer stage any coalition of the receiver and a set of corrupted servers can learn information on the secrets of the sender only from what the corrupted servers may obtain from the sender during the setup stage. Intuitively, this is true because the input and random input of the receiver are independent of the secrets of the sender.

Lemma 5.1.2 *In a one-round* (k, l)-DOT-$\binom{n}{1}$ *protocol, we have that* $H(\mathbf{W}|\mathbf{I}, \mathbf{Y}, \mathbf{D}_T) = H(\mathbf{W}|\mathbf{D}_T)$ *for every* $T \subseteq [l]$.

Proof: Due to (2.2), we have $H(\mathbf{W}|\mathbf{Y}, \mathbf{D}_T) - H(\mathbf{W}|\mathbf{I}, \mathbf{Y}, \mathbf{D}_T) = H(\mathbf{I}|\mathbf{Y}, \mathbf{D}_T) - H(\mathbf{I}|\mathbf{W}, \mathbf{Y},\ \mathbf{D}_T) = 0$. Therefore, $H(\mathbf{W}|\mathbf{I}, \mathbf{Y}, \mathbf{D}_T) = H(\mathbf{W}|\mathbf{Y}, \mathbf{D}_T)$. It suffices to show $H(\mathbf{W}|\mathbf{Y}, \mathbf{D}_T) = H(\mathbf{W}|\ \mathbf{D}_T)$. However, this is equivalent to show that $H(\mathbf{Y}|\ \mathbf{D}_T) = H(\mathbf{Y}|\mathbf{D}_T, \mathbf{W})$. Due to (2.4) and (2.1), we have $H(\mathbf{Y}) \geq H(\mathbf{Y}|\mathbf{D}_T) \geq H(\mathbf{Y}|\mathbf{D}_T, \mathbf{W}) \geq H(\mathbf{Y}|\mathbf{W}, \mathbf{X}) = H(\mathbf{Y})$, which gives the expected equality. □

The last equality shows that given the transcript of the communication between the receiver and the servers labeled by a k-subset $K \subseteq [l]$, any coalition of the receiver and a set of corrupted servers labeled by a t-subset $T \subseteq [l]$ can learn more information on the secrets of the sender only from the messages that $\mathcal{S}_T$ may obtain from the sender and the answers that the receiver may obtain from the servers $\mathcal{S}_{K\setminus T}$. Intuitively, this is true because the servers are assumed deterministic and the answers of the servers $\mathcal{S}_{K\cap T}$ could have been computed by themselves using the messages from both the sender and the receiver.

Lemma 5.1.3 *In a one-round* (k,l)-DOT-$\binom{n}{1}$ *protocol, we have that* $H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{C}_K,\mathbf{D}_T,\mathbf{W}_i) = H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{A}_{K\setminus T},\mathbf{D}_T)$ *and* $H(\mathbf{W}_i|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{A}_{K\setminus T},\mathbf{D}_T) = 0$ *for every* $i\in[n], Y$ *and* $K,T\subseteq[l]$ *where* $|K|=k$.

Proof: Due to (2.7), we have that $H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{C}_K,\mathbf{D}_T,\mathbf{W}_i) = H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{C}_K,\mathbf{D}_T)$. Due to (2.6), we have that $H(\mathbf{A}_{K\cap T}|\mathbf{Q}_{K\cap T},\mathbf{D}_{K\cap T})=0$ and therefore $H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{C}_K,\mathbf{D}_T) = H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{Q}_K,\mathbf{A}_{K\setminus T},\mathbf{D}_T)$. Due to (2.5), we have $H(\mathbf{W}|\ \mathbf{I}=i,\mathbf{Y}=Y,\mathbf{Q}_K,\mathbf{A}_{K\setminus T},\mathbf{D}_T) = H(\mathbf{W}|\mathbf{I}=i,\mathbf{Y}=Y,\mathbf{A}_{K\setminus T},\mathbf{D}_T)$, which implies the first equality. The second equality can be proved in a similar way. □

5.2 Specific Reduction

In this section, we present a specific reduction from (k,l)-DOT-$\binom{n}{1}$ to polynomial interpolation-based PIR where the receiver is semi-honest.

– each server $\mathcal{S}_h$ has the database W and the user $\mathcal{U}$ has an index $i\in[n]$;

- $\mathcal{U}$: choose $V_1,\ldots,V_t \leftarrow \mathbb{F}_q^m$ and send $Q_h = E(i) + \sum_{l=1}^{t}(\lambda_h)^l V_l$ to $\mathcal{S}_h$;
- $\mathcal{S}_h$: send $P(Q_h)$ to $\mathcal{U}$, where $P(Z_1,\ldots,Z_m) = \sum_{j=1}^{n} W_j \cdot P_j(Z_1,\ldots,Z_m)$;
- $\mathcal{U}$: interpolate $G(\lambda) = P(E(i) + \sum_{l=1}^{t}(\lambda_h)^l V_l)$ and output $G(0)$;

Figure 5-1: Polynomial interpolation-based private information retrieval

Polynomial interpolation-based PIR. Let $\mathcal{S}_1,\ldots,\mathcal{S}_k$ be k servers, each of which has the database $W = W_1,\ldots,W_n$. Let $\mathcal{U}$ be a user who wants to retrieve W_i. Figure 5-1 depicts a t-private k-server polynomial interpolation based PIR protocol. The database W is represented by an m-variate polynomial over a finite field $\mathbb{F}_q$ and the index $i\in[n]$ is represented by a point in $\mathbb{F}_q^m$. The recovery of W_i is done using polynomial interpolation along a low degree curve passing through the point. Formally, a t-private k-server polynomial interpolation-based PIR protocol Π [5, 92] consists of an encoding of the indices $E:[n]\to\mathbb{F}_q^m$ and n multivariate polynomials $P_1,\ldots,P_n\in\mathbb{F}_q[Z_1,\ldots,Z_m]$ of degree $\leq d = \lfloor\frac{k-1}{t}\rfloor$ such that $P_j(E(a)) = \delta_{j,a}$ for all

$j, a \in [n]$, where $q > k$ and $\delta_{j,a}$ is the Kronecker's delta symbol. Each server $\mathcal{S}_h$ is associated with a nonzero field element λ_h and represents W as $P(Z_1, \ldots, Z_m)$, where $\lambda_1, \ldots, \lambda_k$ are distinct. To retrieve W_i, the receiver picks $V_1, \ldots, V_t \leftarrow \mathbb{F}_q^m$ and sends to each server $\mathcal{S}_h$ a query Q_h. The server $\mathcal{S}_h$ replies with $A_h = P(Q_h)$. At last, the receiver interpolates the univariate polynomial $G(\lambda)$ and outputs $G(0)$.

– **Input**: $\mathcal{D}$ has n secrets $W \in \mathbb{F}_q^n$ and $\mathcal{U}$ has an index $i \in [n]$;

– **Setup stage:** The sender $\mathcal{D}$ proceeds as follows.

- choose $n+1$ random polynomials $B_0(\lambda), B_1(\lambda), \ldots, B_n(\lambda)$ from $\mathbb{F}_q[\lambda]$, say $B_0(\lambda) = \sum_{a=0}^{k-1} B_{0,a}\lambda^a$ and $B_j(\lambda) = \sum_{a=0}^{\tau} B_{j,a}\lambda^a$ for every $j \in [n]$ such that $B_{0,0} + B_{j,0} = W_j$;
- represent the n secrets as an $(m+1)$-variate polynomial in $\lambda, Z_1, \ldots, Z_m$, that is, $F(\lambda, Z_1, \ldots, Z_m) = B_0(\lambda) + \sum_{j=1}^{n} B_j(\lambda) P_j(Z_1, \ldots, Z_m)$, where P_j is of degree at most $\lfloor (k - \tau - 1)/t \rfloor$ for every $j \in [n]$;
- for any k-subset $K \subseteq [l]$ and integer $h \in K$, choose a random field element $X_{K,h} \leftarrow \mathbb{F}_q$ such that $\sum_{h \in K} X_{K,h} = 0$;
- send $\{X_{K,h} : h \in K \subseteq [l] \text{ and } |K| = k\}$ and $F(\lambda_h, Z_1, \ldots, Z_m)$ to the server $\mathcal{S}_h$ for every $h \in [l]$;

– **Transfer stage:** In this stage, the receiver contacts k servers in order to learn W_i. Suppose the contacted servers are labeled by integers in $K \subseteq [l]$. The receiver and the contacted servers proceed as follows.

- $\mathcal{U}$: send K and $Q_h = E(i) + \sum_{a=1}^{t} (\lambda_h)^a V_a$ to the server $\mathcal{S}_h$ for every $h \in K$, where $V_1, \ldots, V_t$ are chosen from $\mathbb{F}_q^m$ uniformly;
- $\mathcal{S}_h$: send $A_h = F(\lambda_h, Q_h) \prod_{j \in K \setminus \{h\}} \frac{-\lambda_j}{\lambda_h - \lambda_j} + X_{K,h}$ to $\mathcal{U}$ for every $h \in K$;
- $\mathcal{U}$: output $\sum_{h \in K} A_h$.

Figure 5-2: A specific reduction from DOT to PIR

Specific reduction. Let Π be a t-private $(k-\tau)$-server polynomial interpolation-based PIR protocol. Figure 5-2 depicts our specific reduction from (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ to Π. In the setup stage, the sender picks $n+1$ random polynomials $B_0(\lambda), B_1(\lambda), \ldots, B_n(\lambda)$ over $\mathbb{F}_q$ such that $B_0(0) + B_j(0) = W_j$ for every $j \in [n]$. The $B_0(\lambda)$ is of degree $\leq k-1$ while each $B_j(\lambda)$ is of degree $\leq \tau$. Then the sender represents the n secrets as a multivariate polynomial $F(\lambda, Z_1, \ldots, Z_m)$ which is based on the multivariate polynomial used by Π but quite different. At the end of the setup stage, the sender sends to a partially evaluated form of $F(\lambda, Z_1, \ldots, Z_m)$, namely $F(\lambda_h, Z_1, \ldots, Z_m)$, to server $\mathcal{S}_h$ for every $h \in [l]$. The sender also sends necessary randomness to the servers for later use. During the transfer stage, the receiver contacts k out of the l servers. More precisely, he sends to each contacted server a query that is computed in the same way as in Π. In fact, the query is a point on a randomly chosen low degree curve passing through $E(i)$. Each contacted server simply evaluates its multivariate polynomial at the point it receives and replies with the evaluation. At last, the receiver interpolates a univariate polynomial whose evaluation at 0 is exactly the expected secret.

Lemma 5.2.1 *The following statements hold:*

(1) *The protocol depicted by Figure 5-2 is correct.*

(2) *The protocol depicted by Figure 5-2 satisfies the receiver's privacy.*

Proof: (1) It is easy to see that $\sum_{h\in K} A_h = W_i$ due to Lagrange interpolation. The correctness of the protocol follows. (2) Due to the t-privacy of Shamir's threshold secret sharing scheme, any t servers cannot learn the index of the receiver. The receiver's privacy follows. □

Let $\mathcal{K}$ and $\mathcal{T}$ be the sets of all k-subsets and τ-subsets of $[l]$, respectively. For every $K \in \mathcal{K}$ and $h \in K$, let $\mathbf{X}_{K,h}$ be the random variable describing $X_{K,h}$. For every $h \in [l]$ and $0 \leq j \leq n$, let $\mathbf{B}_j(\lambda_h)$ be the random variable describing $B_j(\lambda_h)$. Then

$$\mathbf{A}_h = \prod_{j\in K\setminus\{h\}} \frac{-\lambda_j}{\lambda_h - \lambda_j}\left(\mathbf{B}_0(\lambda_h) + \sum_{j=1}^{n} \mathbf{B}_j(\lambda_h) P_j(\mathbf{Q}_h)\right) + \mathbf{X}_{K,h}. \tag{5.1}$$

On the other hand, for every integer $h \in [l]$, we define

$$\begin{aligned} \mathbf{X}_{k,h} &= (\mathbf{X}_{K,h} : h \in K \in \mathcal{K}), \qquad \mathbf{X}_{B,h} = (\mathbf{B}_0(\lambda_h), \ldots, \mathbf{B}_n(\lambda_h)), \\ \mathbf{D}_h^* &= (\mathbf{X}_{k,h}, \mathbf{X}_{B,h}), \qquad\qquad\qquad \mathbf{D}_h = (\mathbf{X}_{k,h}, \mathbf{F}_h), \\ \mathbf{F}_h &= \mathbf{B}_0(\lambda_h) + \sum_{j=1}^{n} \mathbf{B}_j(\lambda_h) P_j(Z_1, \ldots, Z_m). \end{aligned} \tag{5.2}$$

Following the notations in Section 2.2.4, we can define $\mathbf{A}_T, \mathbf{X}_{k,T}, \mathbf{X}_{B,T}, \mathbf{D}_T^*, \mathbf{D}_T$, and $\mathbf{F}_T$ for every $T \subseteq [l]$. Clearly, $H(\mathbf{D}_T|\mathbf{D}_T^*) = 0$ for every $T \subseteq [l]$. Let $\mathbf{B} = (\mathbf{B}_{0,0}, \ldots, \mathbf{B}_{n,\tau})$ be a random vector describing the coefficient vector $B = (B_{0,0}, \ldots, B_{n,\tau})$ of the polynomials $B_0(\lambda), \ldots, B_n(\lambda)$. Let

$$\Lambda' = \begin{pmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{k-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{k-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_\tau & \cdots & \lambda_\tau^{k-1} \end{pmatrix}, \Lambda'' = \begin{pmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{\tau} \\ 1 & \lambda_2 & \cdots & \lambda_2^{\tau} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_\tau & \cdots & \lambda_\tau^{\tau} \end{pmatrix}$$

and $\Lambda = \mathrm{diag}(\Lambda', \Lambda'', \ldots, \Lambda'')$, where Λ'' occurs exactly n times.

Lemma 5.2.2 *The protocol depicted by Figure 5-2 satisfies the sender's privacy I.*

Proof: W.l.o.g, suppose that $\mathcal{S}_T$ is the set of servers colluding with $\mathcal{U}$, where $T = [\tau]$. Due to Lemma 5.1.2, it suffices to show $H(\mathbf{W}|\mathbf{D}_T) = H(\mathbf{W})$. On one hand, we have $H(\mathbf{W}|\mathbf{D}_T^*) = H(\mathbf{W}|\mathbf{D}_T,\ \mathbf{D}_T^*) \leq H(\mathbf{W}|\mathbf{D}_T) \leq H(\mathbf{W})$. On the other hand, we have that $H(\mathbf{W}|\mathbf{X}_{B,T}) - H(\mathbf{W}|\ \mathbf{D}_T^*) = H(\mathbf{X}_{k,T}|\mathbf{X}_{B,T}) - H(\mathbf{X}_{k,T}|\mathbf{X}_{B,T}, \mathbf{W}) = 0$ in the specific reduction. It suffices to show that $H(\mathbf{W}|\mathbf{X}_{B,T}) = H(\mathbf{W})$.

Let $W \in \mathbb{F}_q^n$ and $X_{B,T} = (B_0(\lambda_1), \ldots, B_n(\lambda_\tau)) \in \mathbb{F}_q^{(n+1)\tau}$. For every $j \in [n]$, let $\rho^j = (1, 0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}_q^{k+n(\tau+1)}$ be a 0-1 vector with support $\{1, k + (j-1)\tau + 1\}$. Consider the following equations with B, the coefficients of the polynomials

$B_0(\lambda), \ldots, B_n(\lambda)$, as unknowns:

$$(1)\ \ \Lambda \cdot B = \begin{pmatrix} B_0(\lambda_1) \\ \vdots \\ B_n(\lambda_\tau) \end{pmatrix} \ \text{ and } \ (2)\ \ \boldsymbol{\rho} \cdot B = \begin{pmatrix} \rho^1 \\ \vdots \\ \rho^n \end{pmatrix} \cdot B = \begin{pmatrix} W_1 \\ \vdots \\ W_n \end{pmatrix}.$$

Clearly, the matrix $\begin{pmatrix} \Lambda \\ \boldsymbol{\rho} \end{pmatrix}$ is of full rank and therefore (1)-(2) has exactly $q^{k-\tau}$ solutions. On the other hand, $\boldsymbol{\rho}$ is of full rank and (2) has exactly $q^{k+n\tau}$ solutions. It follows that

$$\Pr[\mathbf{X}_{B,T} = X_{B,T} | \mathbf{W} = W] = q^{-(n+1)\tau},$$

i.e., $\mathbf{X}_{B,T}$ is uniformly distributed given $\mathbf{W} = W$. It follows that $\mathbf{W}$ and $\mathbf{X}_{B,T}$ are independent and $H(\mathbf{W}|\mathbf{X}_{B,T}) = H(\mathbf{W})$. □

Lemma 5.2.3 *The protocol depicted by Figure 5-2 satisfies the sender's privacy II.*

Proof: Let $i \in [n], K \in \mathcal{K}$ and w.l.o.g $T = [\tau]$. Let $Y = (V_a : a \in [t])$ be the random vectors chosen by $\mathcal{U}$. Due to Lemma 5.1.3, it suffices to show that $H(\mathbf{W}|\mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T}, \mathbf{D}_T) = H(\mathbf{W}|\mathbf{W}_i)$. Let $\Gamma = [n] \setminus \{i\}$. We have that $H(\mathbf{W}|\ \mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T},\ \mathbf{D}_T) = H(\mathbf{W}_\Gamma|\mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T}, \mathbf{D}_T)$ due to Lemma 5.1.3 and $H(\mathbf{W}|\mathbf{W}_i) = H(\mathbf{W}_\Gamma)$ due to Equality (2.3). Thus, it suffices to show that $H(\mathbf{W}_\Gamma|\mathbf{I} = i, \mathbf{Y} = Y,\ \mathbf{A}_{K\setminus T}, \mathbf{D}_T) = H(\mathbf{W}_\Gamma)$. Due to $H(\mathbf{D}_T|\mathbf{D}^*_T) = 0$, we have that $H(\mathbf{W}_\Gamma) \geq H(\mathbf{W}_\Gamma|\mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T}, \mathbf{D}_T) \geq H(\mathbf{W}_\Gamma|\mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T}, \mathbf{D}^*_T)$. Hence, it is enough to show that $H(\mathbf{W}_\Gamma|\mathbf{I} = i, \mathbf{Y} = Y,\ \mathbf{A}_{K\setminus T}, \mathbf{D}^*_T) = H(\mathbf{W}_\Gamma)$.

Let $W_\Gamma \in \mathbb{F}_q^{n-1}$ and $(A_{K\setminus T}, D^*_T)$ be such that $\Pr[\mathbf{W}_\Gamma = W_\Gamma] > 0$ and $\Pr[\mathbf{A}_{K\setminus T} = A_{K\setminus T}, \mathbf{D}^*_T = D^*_T|\mathbf{W}_\Gamma = W_\Gamma, \mathbf{I} = i, \mathbf{Y} = Y] > 0$. We want to compute

$$p_1 = \Pr[\ \mathbf{A}_{K\setminus T} = A_{K\setminus T}, \mathbf{D}^*_T = D^*_T|\mathbf{W}_\Gamma = W_\Gamma, \mathbf{I} = i, \mathbf{Y} = Y]. \tag{5.3}$$

Clearly, $\mathbf{X}_{k,T}$ is independent of $\mathbf{W}_\Gamma$ given $\mathbf{I} = i$ and $\mathbf{Y} = Y$. On the other hand, the messages $X_{k,T}$ consists of $\tau\binom{l-1}{k-1}$ random field elements. Hence, we have that

$$\Pr[\mathbf{X}_{k,T} = X_{k,T} | W_\Gamma, i, Y] = q^{-\tau\binom{l-1}{k-1}}. \tag{5.4}$$

Due to Lemma 5.1.3, we have that $H(\mathbf{W}_i | \mathbf{I} = i, \mathbf{Y} = Y, \mathbf{A}_{K\setminus T},\ \mathbf{D}_T^*) = 0$. Therefore, $(A_{K\setminus T}, D_T^*)$ uniquely determines a secret $W_i \in \mathbb{F}_q$ given $\mathbf{I} = i$ and $\mathbf{Y} = Y$. Due to Equality (2.3) and the choice of $X_{k,T}$ in the specific reduction, $\mathbf{W}_i$ is independent of $(\mathbf{X}_{k,T}, \mathbf{W}_\Gamma)$ given $\mathbf{I} = i$ and $\mathbf{Y} = Y$. In other words, we have that

$$\Pr[\mathbf{W}_i = W_i | X_{k,T}, W_\Gamma, i, Y] = \Pr[\mathbf{W}_i = W_i]. \tag{5.5}$$

$X_{B,T}$ is consistent with W and independent of $(\mathbf{X}_{k,T}, \mathbf{I}, \mathbf{Y})$. Therefore, as in the proof of Lemma 5.2.2, we have that

$$\Pr[\mathbf{X}_{B,T} = X_{B,T} | W_i, X_{k,T}, W_\Gamma, i, Y] = \Pr[\mathbf{X}_{B,T} = X_{B,T} | W_i, W_\Gamma] = q^{-(n+1)\tau}. \tag{5.6}$$

Due to Equalities (2.5), (5.2) and (5.1), the tuple $(X_{B,T}, W_i, X_{k,T}, W_\Gamma, i, Y)$ uniquely determines A_T. Clearly, any valid answers $A_{K\setminus T}$ replied by servers $\mathcal{S}_{K\setminus T}$ should satisfy $\sum_{h\in K\setminus T} A_h + \sum_{h\in T} A_h = W_i$. Therefore, the domain of $\mathbf{A}_{K\setminus T}$ is equal to $\mathbb{A}_{K\setminus T} = \{(a_1, \ldots, a_{|K\setminus T|}) : \sum_{h=1}^{|K\setminus T|} a_h = W_i - \sum_{h\in T} A_h\}$. Furthermore, $\mathbf{A}_{K\setminus T}$ is uniformly distributed on $\mathbb{A}_{K\setminus T}$ (see the third step of the transfer stage). Hence,

$$\Pr[\mathbf{A}_{K\setminus T} = A_{K\setminus T} | X_{B,T}, W_i, X_{k,T}, W_\Gamma, i, Y] = |\mathbb{A}_{K\setminus T}|^{-1} = q^{-(|K\setminus T|-1)}. \tag{5.7}$$

Due to Equalities (5.3), (5.4), (5.5), (5.6) and (5.7), we have that $p_1 = q^{-\tau\binom{l-1}{k-1}} \cdot \Pr[\mathbf{W}_i = W_i] \cdot q^{-(n+1)\tau} \cdot q^{-(|K\setminus T|-1)}$. Hence, $(\mathbf{A}_{K\setminus T}, \mathbf{D}_T^*)$ and $\mathbf{W}_\Gamma$ are independent given $(\mathbf{I}, \mathbf{Y}) = (i, Y)$. Hence, $H(\mathbf{W}_\Gamma | \mathbf{I} = i, \mathbf{Y} = Y,\ \mathbf{A}_{K\setminus T}, \mathbf{D}_T^*) = H(\mathbf{W}_\Gamma)$. □

The communication complexity between the receiver and servers is $\gamma(n) = O(m)$. The PIR depicted by Figure 5-1 requires $m = O\left(n^{1/\lfloor \frac{k-\tau-1}{t} \rfloor}\right)$. Hence, we have

Theorem 5.2.1 *Let t, τ, k, l be positive integers such that $l \geq k \geq t+\tau+1$. The spe-*

cific reduction yields a (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ protocol of communication complexity $O\left(n^{1/\lfloor\frac{k-\tau-1}{t}\rfloor}\right)$ between a semi-honest receiver and servers.

– **Input:** $\mathcal{D}$ has n secrets $W \in \mathbb{F}_q^n$ and $\mathcal{U}$ has an index $i \in [n]$;

– **Subroutine:** $\Pi = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$, a t-private $(k-\tau)$-server SPIR protocol;

– **Setup stage:** The sender invokes the $\text{CNF}_{\tau,k}$ with each k-subset of the l servers in order to distribute the secrets. Therefore, the sender invokes exactly $\binom{l}{k}$ $\text{CNF}_{\tau,k}$ instances, where each of them is labeled by a k-subset $K \in \mathcal{K}$. We only describe one (labeled by K) of these instances. The sender $\mathcal{D}$ proceeds as follows.

- choose $W_{K,T} \leftarrow \mathbb{F}_q^n$ such that $\sum_{T\subseteq K,|T|=\tau} W_{K,T} = W$;
- send $D_h = \{W_{K,T} : T \subseteq K, |T| = \tau, h \in K \setminus T\}$ to $\mathcal{S}_h$ for $h \in K$;
- send a random string $X = \{X_{K,T} : T \subseteq K, |T| = \tau\}$ to $\mathcal{S}_h$ for $h \in K$;

– **Transfer stage:** The user contact k servers that are labeled by integers in the k-subset $K \subseteq [l]$. Specifically, for every τ-subset $T \subseteq K$, the user invokes an SPIR instance, where the database is $W_{K,T}$. Below, we only describe one of the $\binom{k}{\tau}$ SPIR instances. We suppose the user is contacting the servers labeled by $K \setminus T$ in the described instance.

- $\mathcal{U}$: picks a random string $Y_{K,T}$ for the instance;
- $\mathcal{U}$: send $Q_{K,T,h} = \mathcal{Q}(n, i, Y_{K,T})_h$ to $\mathcal{S}_h$ for every $h \in K \setminus T$;
- $\mathcal{S}_h$: send $A_{K,T,h} = \mathcal{A}(n, W_{K,T}, Q_{K,T,h}, X_{K,T})$ to $\mathcal{U}$, where $h \in K \setminus T$;
- $\mathcal{U}$: compute $\Phi_{K,T} = \mathcal{R}(i, Y_{K,T}, \{Q_{K,T,h}, A_{K,T,h} : h \in K \setminus T\})$;

At last, the user have computed $\binom{k}{\tau}$ intermediate results. The final output is defined to be the sum of these results, i.e., $\sum_{T:T\in\mathcal{T},T\subseteq K} \Phi_{K,T}$.

Figure 5-3: A general reduction from DOT to SPIR

5.3 General Reduction

In this section, we transform any PIR to communication-efficient DOT. The core of our transformation is a reduction from DOT to SPIR which is depicted by Figure 5-3.

During the setup stage, the sender $\mathcal{D}$ distributes the secrets among each k-subset of the l servers using $\text{CNF}_{\tau,k}$. Clearly, any τ servers may be participants in many instances of $\text{CNF}_{\tau,k}$. However, in each instance, they miss at least one share and therefore cannot learn any information about the sender's secrets. During the transfer stage, the receiver contacts k servers. He invokes an SPIR instance with each $(k-\tau)$-subset of the contacted servers. In each SPIR instance, the receiver retrieves the ith component of the $\text{CNF}_{\tau,k}$ share held by the $(k-\tau)$ servers participating the SPIR instance. At last, the receiver learns the ith components of the $\binom{k}{k-\tau}$ $\text{CNF}_{\tau,k}$ shares. Clearly, these components sum to the expected secret, i.e., W_i.

Lemma 5.3.1 *The following statements hold.*

(1) *The protocol depicted by 5-3 is correct.*

(2) *The protocol depicted by 5-3 satisfies the receiver's privacy.*

(3) *The protocol depicted by 5-3 satisfies the sender's privacy I.*

Proof: (1) The correctness of the protocol follows from the correctness of the SPIR protocol. (2) The receiver's privacy follows from the t-privacy of the SPIR protocol. (3) The sender's privacy I follows from the τ-privacy of $\text{CNF}_{\tau,k}$. □

Lemma 5.3.2 *The protocol depicted by 5-3 satisfies the sender's privacy II.*

Proof: Let $\mathcal{S}_K$ and $\mathcal{S}_{T'}$ be the servers contacted by $\mathcal{U}$ and colluding with $\mathcal{U}$ respectively, where $K \in \mathcal{K}$ and $T' \in \mathcal{T}$. Let $\mathcal{T}^*$ be the set of τ-subsets of K, $\mathcal{T}_1^* = \{T \in \mathcal{T}^* : (T' \cap K) \subseteq T\}$ and $\mathcal{T}_2^* = \mathcal{T}^* \setminus \mathcal{T}_1^*$. Let $\mathbf{A}_{K,\mathcal{T}_1^*} = (\mathbf{A}_{K,T,K\setminus T} : T \in \mathcal{T}_1^*)$ and $\mathbf{A}_{K,\mathcal{T}_2^*} = (\mathbf{A}_{K,T,K\setminus T} : T \in \mathcal{T}_2^*)$, where $\mathbf{A}_{K,T,K\setminus T} = (\mathbf{A}_{K,T,h} : h \in K \setminus T)$ for every $T \in \mathcal{T}^*$. For every $T' \subseteq [l]$, we set $\mathbf{D}_{T'} = (\mathbf{W}_{K,T} : T \in \mathcal{T}_2^*)$.

Semi-honest case. Suppose that the receiver is semi-honest. Given $i \in [n]$ and Y_K, we have that $H(\mathbf{A}_{K,\mathcal{T}_2^*}|\mathbf{I} = i, \mathbf{Y}_K = Y_K, \mathbf{D}_{T'}) = 0$ due to (2.6) and (2.5). By this equality and Lemma 5.1.3, it is sufficient to show that $H(\mathbf{W}|\mathbf{I} = i, \mathbf{Y}_K = Y_K, \mathbf{A}_{K,\mathcal{T}_1^*}, \mathbf{D}_{T'}) = H(\mathbf{W}|\mathbf{W}_i)$.

Let $A_{K,\mathcal{T}_1^*}$ and $D_{T'}$ be in the domains of $\mathbf{A}_{K,\mathcal{T}_1^*}$ and $\mathbf{D}_{T'}$, where $D_{T'} = (W^\star_{K,T} : T \in \mathcal{T}_2^*)$. Clearly, the ith secret (say b) can be determined by $A_{K,\mathcal{T}_1^*}$ and $D_{T'}$ given $(\mathbf{I}, \mathbf{Y}_K) = (i, Y_K)$. Let $W, W' \in \mathbb{F}_q^n$ be such that $W_i = W'_i = b$. Due to the τ-privacy of $\mathrm{CNF}_{\tau,k}$, we have that

$$\Pr\left[\mathbf{D}_{T'} = D_{T'}|W, i, Y_K\right] = \Pr\left[\mathbf{D}_{T'} = D_{T'}|W', i, Y_K\right]. \tag{5.8}$$

Let $\delta = |\mathcal{T}_1^*|$. For $S = W$ or W', we define $\Omega_S = \{(U_1, \ldots, U_\delta) \in (\mathbb{F}_q^n)^\delta : \sum_{h=1}^{\delta} U_h = S - \sum_{T\in\mathcal{T}_2^*} W^\star_{K,T}\}$. For every $(b_1, \ldots, b_\delta) \in \mathbb{F}_q^\delta$ such that $\sum_{h=1}^{\delta} b_h = b - \sum_{T\in\mathcal{T}_2^*}[W^\star_{K,T}]_i$ and $V \in \Omega_W, V' \in \Omega_{W'}$ such that $V_i = V'_i = (b_1, \ldots, b_\delta)$, it is easy to see that

$$\Pr[\mathbf{W}_{K,\mathcal{T}_1^*} = V|D_{T'}, W, i, Y_K] = \Pr[\mathbf{W}_{K,\mathcal{T}_1^*} = V'|D_{T'}, W', i, Y_K]. \tag{5.9}$$

For every $T \in \mathcal{T}^*$, an SPIR is executed by $\mathcal{U}$ and $\mathcal{S}_{K\setminus T}$. Due to the data privacy of the SPIR, we have that

$$\Pr[\mathbf{A}_{K,T,K\setminus T} = A_{K,T,K\setminus T}|W_{K,T}, i, Y_{K,T}] = \Pr[\mathbf{A}_{K,T,K\setminus T} = A_{K,T,K\setminus T}|W'_{K,T}, i, Y_{K,T}]. \tag{5.10}$$

for any n-tuples of secrets $W_{K,T}$ and $W'_{K,T}$ such that $[W'_{K,T}]_i = [W_{K,T}]_i$, and any answer tuple $A_{K,T}$ received by $\mathcal{U}$. Due to (5.10), we have that

$$\begin{aligned}\Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}|\mathbf{W}_{K,\mathcal{T}_1^*} = V, D_{T'}, W, i, Y_K] \\ = \Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}|\mathbf{W}_{K,\mathcal{T}_1^*} = V', D_{T'}, W', i, Y_K].\end{aligned} \tag{5.11}$$

Equality (5.9) and (5.11) jointly imply $\Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}|D_{T'}, W, i, Y_K] = \Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}|D_{T'}, W', i, Y_K]$. This equality and (5.8) jointly imply the following equality: $\Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}, \mathbf{D}_{T'} = D_{T'}|W, i, Y_K] = \Pr[\mathbf{A}_{K,\mathcal{T}_1^*} = A_{K,\mathcal{T}_1^*}, \mathbf{D}_{T'} = D_{T'}|W', i, Y_K]$.

Hence, $(\mathbf{A}_{K,\mathcal{T}_1^*}, \mathbf{D}_{T'})$ and $\mathbf{W}$ are independent given $(\mathbf{I}, \mathbf{Y}_K, \mathbf{W}_i) = (i, Y_K, b)$. The expected equality holds and therefore the sender's privacy II is satisfied.

Malicious case. In order to learn as much information as possible, T' should be a subset of K, i.e., $T \in \mathcal{T}^*$. Due to the τ-privacy of $\mathrm{CNF}_{\tau,k}$, the coalition of $\bar{\mathcal{U}}$ and $\mathcal{S}_{T'}$ is able to learn W as long as the share $W_{K,T'}$ is known. Consider the SPIR executed by $\bar{\mathcal{U}}$ and $\mathcal{S}_{K\setminus T'}$. If $\bar{\mathcal{U}}$ contributes no valid index, then the coalition learns no information on $W_{K,T'}$ and therefore no information on W. If $\bar{\mathcal{U}}$ contributes an index $j \in [n]$, then the coalition learns at most W_j due to the data privacy of the SPIR and the τ-privacy of $\mathrm{CNF}_{\tau,k}$. □

The receiver invokes $\binom{k}{\tau}$ instances of $\mathcal{P}$ with the interacted servers. Let $\gamma(n)$ be the communication complexity of $\mathcal{P}$. Then the communication complexity between the receiver and servers is $\binom{k}{\tau}\gamma(n)$.

Theorem 5.3.1 *If there is a t-private $(k-\tau)$-server* SPIR *protocol of communication complexity $\gamma(n)$, then for every integer $l \geq k$, there is a (t,τ)-private (k,l)-*DOT-$\binom{n}{1}$ *protocol of communication complexity $O(\gamma(n))$ between the receiver and servers.*

Gertner et al. [42] proposed a transformation from any one-round t-private k-server PIR protocol of communication complexity $\gamma(n)$ to a one-round t-private $(k+t)$-server SPIR protocol of communication complexity $O(\gamma(n))$. Due to this transformation and Theorem 5.3.1, we have

Theorem 5.3.2 *If there is a t-private $(k-t-\tau)$-server* PIR *protocol of communication complexity $\gamma(n)$, then for every integer $l \geq k$, there is a (t,τ)-private (k,l)-*DOT-$\binom{n}{1}$ *protocol of communication complexity $O(\gamma(n))$ between the receiver and servers.*

Applying Theorem 5.3.2 to the polynomial interpolation-based t-private k-server PIR of [5, 92] gives:

Corollary 5.3.1 *For integers t, τ, k, l such that $1 \leq t \leq k-2$, $0 \leq \tau \leq k-1-t$ and $\frac{3t+1}{2} + \tau \leq k \leq l$, there is a (t,τ)-private (k,l)-*DOT-$\binom{n}{1}$ *protocol of communication complexity $O\left(n^{1/\lfloor \frac{2(k-t-\tau)-1}{t} \rfloor}\right)$ between the receiver and servers.*

Note that it is required that $k \geq \frac{3t+1}{2} + \tau$ for obtaining sublinear communication complexity between the receiver and servers in Corollary 5.3.1.

The recent progress [96, 36, 55, 22] in LDCs research yields 1-private 3-server PIR protocols of communication complexity $\exp(O(\sqrt{\log n \log\log n}))$. Barkol et al. [2] proposed a transformation from any 1-private k-server PIR of communication complexity $\gamma(n)$ to t-private k^t-server PIR of communication complexity $O(\gamma(n))$. Due to these results and Theorem 5.3.2, we have

Corollary 5.3.2 *For integers t, τ, l such that $l \geq 3^t + t + \tau$, there is a (t,τ)-private $(3^t+t+\tau, l)$-DOT-$\binom{n}{1}$ protocol of communication complexity $\exp(O(\sqrt{\log n \log\log n}))$ between the receiver and servers.*

Let $t = \tau = 1$ and $l \geq 5$. We have a $(1,1)$-private $(5,l)$-DOT-$\binom{n}{1}$ protocol of communication complexity $\exp(O(\sqrt{\log n \log\log n}))$ between the receiver and servers.

5.4 Performance of The Reductions

In this section, we evaluate our specific reduction and general reduction.

Privacy: The one-round (k,l)-DOT-$\binom{n}{1}$ protocols of [11] achieve $(k-1)$-receiver privacy, $(k-1)$-sender's privacy I and 0-sender's privacy II. For semi-honest receivers, if we take $t = k-1$ and $\tau = 0$ in the specific reduction and apply the encoding E of indices of [92] for $m = n$, then the one-round (k,l)-DOT-$\binom{n}{1}$ protocols of [11] can be obtained. The (k,l)-DOT-$\binom{n}{1}$ protocols of [73] achieve t-receiver privacy, $(k-1-t)$-sender's privacy I and $(k-1-t)$-sender's privacy II for any positive integer $t < k-1$. Our specific reduction yields (k,l)-DOT-$\binom{n}{1}$ achieving the same privacy whenever $t + \tau = k-1$. Hence, the specific reduction shows a tradeoff between receiver-server communication complexity and privacy.

The specific reduction is only proved for semi-honest receivers while the general reduction is fully secure. It is left open to modify the specific reduction such that the resulting DOT is secure even in the presence of malicious receivers.

Communication complexity: Let t, τ be fixed positive integers. It has been proved by [73] that in any (t,τ)-private (k,l)-DOT-$\binom{n}{1}$, we have that $k \geq t+\tau+1$. If

$t+\tau+1 \leq k \leq 2t+\tau$, then the (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ obtained by the general reduction require linear communication complexity between the receiver and servers due to Corollary 5.3.1. If $k > 2t+\tau$, then $\left\lfloor \frac{2(k-t-\tau)-1}{t} \right\rfloor > \left\lfloor \frac{k-\tau-1}{t} \right\rfloor$ and therefore the (t,τ)-private (k,l)-DOT-$\binom{n}{1}$ obtained by the general reduction is more efficient due to Theorem 5.2.1 and Corollary 5.3.1. Furthermore, for small values of t and τ, the reduction given by Corollary 5.3.2 is much better than Theorem 5.2.1 and Corollary 5.3.1 since it is based on the 3-server PIR protocols of extremely small communication complexity $\exp(O(\sqrt{\log n \log\log n}))$.

Availability: For the consideration of availability, the receiver should interact with a small number of servers. The specific reduction of Theorem 5.2.1 achieves the lower bound of [73] and therefore is optimal. In contrast, the general reductions of Corollary 5.3.1 and 5.3.2 are not optimal because they require $k \geq \frac{3t+1}{2} + \tau$ and $k = 3^t + t + \tau$, respectively.

Setup complexity: The *setup complexity* of our reductions could be large. For example, in the setup stage of our specific reduction, each server receives $\binom{l-1}{k-1}$ field elements and an m-variate polynomial of small constant degree; in the setup stage of our general reduction, each server receives $\binom{l-1}{k-1}\binom{k-1}{t}n + \binom{l}{k}\binom{k}{t}$ field elements. However, the setup complexity is reasonable if $l = k$ and t is a constant less than k. DOT protocols in a setting where $l = k$ have been considered by [25]. Our reductions are efficient for that setting.

Independence of secrets: The assumption that the sender's secrets are *totally independent* has been made in previous sections. However, our security proofs for the general reduction do not depend on this assumption.

Chapter 6

Oblivious Transfer and n-Variate Linear Function Evaluation

In this chapter, we present an unconditionally secure reduction from $\binom{n}{1}$-OT to $\mathcal{C}$-OLFE$_n$, where $\mathcal{C}$ contains all unit vectors of length n. It is easy to note that the reduction is trivial when $\mathcal{C}$ is equal to the set of all unit vectors of length n. The reduction becomes nontrivial and interesting when $\mathcal{C}$ is strictly larger. This is true because a cheating receiver Bob is able to learn many linear combinations of the secrets of the sender Alice, i.e., he is not restricted to learn a single secret. We focus on the nontrivial case in this chapter.

6.1 The Reduction from OT to OLFE

Notations: For a positive integer n, we denote by $\mathbb{S}_n$ the set of all permutations of integers in $[n]$. The complement of a set P is denoted by $\bar{P}$. Let M be a $k \times n$ matrix. For $P \subseteq [k]$ and $Q \subseteq [n]$, the submatrix of M with rows indexed by P and columns indexed by Q is denoted by $M[\![P, Q]\!]$; specifically, $M[\![P, *]\!] = M[\![P, [n]]\!]$. The *support* of a vector v is defined to be $N(v) = \{t : v_t \neq 0\}$. $\mathbf{1}$ and $\mathbf{0}$ denote the all-one and all-zero vectors, respectively.

Let $f = \binom{n}{1}$-OT and $g = \mathcal{C}$-OLFE$_n$. We present a reduction from f to g in this section. As a basic step, we suppose $g = \mathbb{F}^n$-OLFE$_n$ and have a reduction depicted

by Figure 6-1. At the beginning of the reduction, the sender Alice picks k random permutations $\phi_1, \ldots, \phi_k$ of the integers in $[n]$ and k random matrices $X_1, \ldots, X_k$ of order n such that the ith columns of the k matrices together with the random permutations $\phi_1, \ldots, \phi_k$ determine exactly one secret, i.e., the secret s_i, where $i \in [n]$. More precisely, s_i is additively shared as the sum of k random field elements which residue in the ith columns of the k matrices. Specifically, the jth share of s_i is the $(\phi_j(i), i)$ entry of the matrix X_j for every $j \in [k]$. Then the sender and the receiver invoke the trusted third party $\mathcal{T}^g$ that implements the functionality of g. The sender simply feeds $\mathcal{T}^g$ with the rows of the k random matrices and the receiver provides his choice $c \in [n]$ to $\mathcal{T}^g$. If the receiver is honest, then he will obtain the cth columns of the k random matrices. After the invocations of $\mathcal{T}^g$, the sender reveals the k random permutations to the receiver. Clearly, these permutations allow the receiver to locate the k shares of s_c and therefore sum the shares in order to learn s_c.

- input: Alice has n secrets $s \in \mathbb{F}^n$ and Bob has a choice $c \in [n]$;
- subroutine: the trusted third party $\mathcal{T}^g$;

1. Alice: choose $\phi_j \leftarrow \mathbb{S}_n$ and $X_j \leftarrow \mathbb{F}^{n \times n}$ for every $j \in [k]$ such that $\sum_{j=1}^{k} X_j[\![\phi_j(i), i]\!] = s_i$ for every $i \in [n]$;
2. $\mathcal{T}^g$: for $(j, i) \in [k] \times [n]$, Alice and Bob proceeds as follows
   - Alice: send $(X_j[\![i, 1]\!], \ldots, X_j[\![i, n]\!])$ to $\mathcal{T}^g$;
   - Bob: send the cth unit vector in $\mathbb{F}^n$ to $\mathcal{T}^g$ and receive Y_{ji} from $\mathcal{T}^g$;
3. Alice: send the permutations $\phi_1, \ldots, \phi_k$ to Bob;
4. Bob: output $\sum_{j=1}^{k} Y_{j\phi_j(c)}$.

Figure 6-1: A reduction from $\binom{n}{1}$-OT to $\mathcal{C}$-OLFE$_n$ (π^g)

The correctness of the reduction is clear from our description above and formally proved in the following lemma.

Lemma 6.1.1 *If Alice and Bob are honest, then* $\mathrm{IDEAL}_{f,\mathcal{S}} = \mathrm{EXEC}_{\pi^g,\mathcal{H}}$*, where* $\mathcal{S}$ *is the ideal model adversary corrupting no party and* $\mathcal{H}$ *is the* g*-hybrid model adversary corrupting no party.*

Proof: For every (k, s, c, z), we have that $\mathrm{IDEAL}_{f,\mathcal{S}}(k, s, c, z) = (\perp, \perp, s_c) \equiv (\perp, \perp, \sum_{j=1}^{k} X_j[\![\phi_j(c), c]\!]) = (\perp, \perp, \sum_{j=1}^{k} Y_{j\phi_j(c)}) = \mathrm{EXEC}_{\pi^g,\mathcal{H}}(k, s, c, z)$, where the random variables depend on the uniform and independent coin tosses of Alice and Bob. □

- input: (k, s, z), where $s \in \mathbb{F}^n$ is Alice's input and z is an auxiliary input;
- subroutine: the g-hybrid model adversary $\mathcal{H}$ and the trusted third party $\mathcal{T}^f$.

1. feed $\mathcal{H}$ with (k, s, z);
2. receive the k matrices $X' = (X'_1, \ldots, X'_k)$ which are decided by $\mathcal{H}$ and then sent to $\mathcal{T}^g$ by Alice.
3. receive the k permutations $\phi' = (\phi'_1, \ldots, \phi'_k)$ which are decided by $\mathcal{H}$ and then sent to Bob by Alice.
4. for every $i \in [n]$, set $s'_i = \sum_{j=1}^{k} X'_j[\![\phi'_j(i), i]\!]$ and send $s' = (s'_1, \ldots, s'_n)$ to $\mathcal{T}$.
5. output whatever $\mathcal{H}$ outputs, say $\mathcal{H}(k, s, z, \phi', X')$.

Figure 6-2: Ideal model adversary corrupting Alice for π^g

We observe that the receiver never sends any messages to the sender. Therefore, the sender cannot learn even any partial information about the choice c. This observation gives us an intuition for the proof of the receiver's privacy below.

Lemma 6.1.2 *For any* g*-hybrid model adversary* $\mathcal{H}$ *corrupting Alice, there is an ideal model adversary* $\mathcal{S}$ *corrupting Alice whose running time is polynomial in that of* $\mathcal{H}$ *such that* $\mathrm{IDEAL}_{f,\mathcal{S}} \equiv \mathrm{EXEC}_{\pi^g,\mathcal{H}}$.

Proof: The ideal model adversary $\mathcal{S}$ is depicted by Figure 6-2. For every (k, s, c, z), we have that $\mathrm{IDEAL}_{f,\mathcal{S}}(k, s, c, z) = (\mathcal{H}(k, s, z, \phi', X'), \perp, s'_c) = (\mathcal{H}(k, s, z, \phi', X'), \perp, \sum_{j=1}^{k} X'_j[\![\phi'_j(c), c]\!]) \equiv (\mathcal{H}(k, s, z, \phi, X), \perp, \sum_{j=1}^{k} X_j[\![\phi_j(c), c]\!]) = (\mathcal{H}(k, s, z, \phi, X), \perp$

$, \sum_{j=1}^{k} Y_{j\phi_j(c)}) = \mathrm{EXEC}_{\pi^g,\mathcal{H}}(k, s, c, z)$, where the random variables only depend on the uniform and independent coin tosses of $\mathcal{H}$. $\square$

It remains to show the sender's privacy. A malicious receiver may not send the real choice c to $\mathcal{T}^g$ throughout. Instead, he may send an arbitrary combination of the vectors in $\mathbb{F}_q^n$ during the invocations of $\mathcal{T}^g$ in order to obtain more than one secret. Intuitively, the receiver can learn more than one secret only if he can correctly locate the shares of more than one secrets. However, the permutations are chosen independently and randomly, which makes the success probability of the malicious receiver exponentially small. In the remaining of this section, we give a rigorous and formal proof for this intuition. For any $(k, c) \in \mathbb{N} \times [n]$, the choice vectors sent to $\mathcal{T}^g$ by Bob are actually decided by $\mathcal{H}$ and therefore arbitrary. Let $c_{ji} \in \mathbb{F}^n$ be the choice vector sent by Bob in the (j, i)-th invocation of $\mathcal{T}^g$ for every $(j, i) \in [k] \times [n]$. For any $(k, s) \in \mathbb{N} \times \mathbb{F}^n$, the honest sender Alice always choose k matrices $\{X_j \in \mathbb{F}^{n\times n} : j \in [k]\}$ according to the specification of π^g. For every $(j, i) \in [k] \times [n]$, an answer Y_{ji} is sent to Bob by $\mathcal{T}^g$. Clearly, we have the following system of linear equations:

$$\begin{cases} c_{ji} \cdot X_j[\![i, *]\!] = Y_{ji} & \text{for every } (j, i) \in [k] \times [n]; \\ \sum_{j=1}^{k} X_j[\![\phi_j(i), i]\!] = s_i & \text{for every } i \in [n]. \end{cases} \tag{6.1}$$

In equation system (6.1), the vector of unknowns is $X = \big(X_1[\![1, 1]\!], \ldots, X_1[\![1, n]\!], \ldots, X_1[\![n, n]\!], \ldots, X_k[\![n, n]\!]\big)$ and the vector of constant terms is $Y = \big(Y_{11}, \ldots, Y_{1n}, \ldots, Y_{kn}, s_1, \ldots, s_n\big)$. We define the subsets of indices $P_i \subseteq [kn], Q_i \subseteq [kn^2]$ and $S_{ji} \subseteq [kn^2]$

$$\begin{aligned} P_i &= \big\{(j-1)n + \phi_j(i) : j \in [k]\big\}, \\ Q_i &= \big\{(j-1)n^2 + (\phi_j(i)-1)n + i : j \in [k]\big\}, \\ S_{ji} &= \big\{h : (j-1)n^2 + (i-1)n + 1 \le h \le (j-1)n^2 + in\big\} \subseteq [kn^2]. \end{aligned} \tag{6.2}$$

for every $i \in [n]$ and $(j, i) \in [k] \times [n]$. Let C be the coefficient matrix of (6.1). Let V_{que} and $V_{\mathrm{sec}} \subseteq \mathbb{F}^{kn^2}$ be the vector spaces spanned by the first kn rows and the last n rows of C, respectively.

Lemma 6.1.3 *Let j be taken over $[k]$ and i, h be taken over $[n]$. Then we have that*

- $\{Q_i : i \in [n]\}$ *and* $\{S_{ji} : (j,i) \in [k] \times [n]\}$ *are composed of pairwise disjoint sets;*
- $|Q_h \cap S_{ji}| \leq 1$ *and it is equal to 1 only if* $(j-1)n + i \in P_h$*;*
- $C[\![(j-1)n + i, S_{ji}]\!] = c_{ji}$ *and* $C[\![(j-1)n + i, \bar{S}_{ji}]\!] = \mathbf{0}$*;*
- $C[\![kn + i, Q_i]\!] = \mathbf{1}$ *and* $C[\![kn + i, \bar{Q}_i]\!] = \mathbf{0}$.

Lemma 6.1.4 *Let* $w = \sum_{i=1}^{n} \alpha_i \cdot C[\![kn + i, *]\!] \in V_{\text{sec}}$. *Then* $N(w) = \cup_{i \in N(\alpha)} Q_i$.

Proof: By Lemma 6.1.3, the support of $C[\![kn + i, *]\!]$ is Q_i for every $i \in [n]$ and all Q_i are pairwise disjoint. Hence, $N(w) = \cup_{i \in N(\alpha)} Q_i$. □

Lemma 6.1.5 *Let* $V_{\text{int}} = V_{\text{que}} \cap V_{\text{sec}}$. *If* $\dim(V_{\text{int}}) = m$, *then there is a subset* I *of* $[n]$ *of cardinality* m *such that* V_{int} *is equal to the row space of* $C[\![kn + I, *]\!]$.

Proof: For every $w \in V_{\text{int}}$, there exists $\alpha \in \mathbb{F}^n$ such that $w = \sum_{i=1}^{n} \alpha_i \cdot C[\![kn + i, *]\!]$. Let $I_w = N(\alpha)$ and $I = \cup_{w \in V_{\text{int}}} I_w$. Then w is in the row space of $C[\![kn + I, *]\!]$. For every $h \in I$, let $w \in V_{\text{int}}$ be such that $h \in I_w$. Let $w = \sum_{i=1}^{n} \alpha_i \cdot C[\![kn + i, *]\!]$ for some $\alpha \in \mathbb{F}^n$. Due to Lemma 6.1.4, $N(\alpha) = I_w$ and the support of w is the disjoint union of the supports of $C[\![kn + i, *]\!]$ (i.e. Q_i), where i is taken over I_w. Due to Lemma 6.1.3, Q_i intersects the support of $C[\![(\lambda - 1)n + \tau, *]\!]$ only if $(\lambda - 1)n + \tau \in P_i$. Since $w \in V_{\text{que}}$, $C[\![kn + i, *]\!]$ must be a linear combination of the $C[\![(\lambda - 1)n + \tau, *]\!]$'s, where $(\lambda - 1)n + \tau \in P_i$. Hence, $C[\![kn + i, *]\!] \in V_{\text{que}}$. In particular, $C[\![kn + h, *]\!] \in V_{\text{que}}$. It follows that $C[\![kn + h, *]\!] \in V_{\text{int}}$. Hence, V_{int} is equal to the row space of $C[\![kn + I, *]\!]$ and $|I| = \dim(V_{\text{int}}) = m$. □

The following lemma shows that a dishonest receiver Bob can obtain more than one secret only with probability $\leq 2^{-k}$, which is negligible.

Lemma 6.1.6 *If* $k \geq 2$, *then* $\Pr[2 \leq \dim(V_{\text{int}}) \leq n] \leq 2^{-k}$, *where the probability is taken over the random permutations* $\phi_1, \ldots, \phi_k \leftarrow \mathbb{S}_n$.

Proof: Due to Lemma 6.1.5, for every $2 \leq m \leq n$, $\dim(V_{\text{int}}) = m$ if and only if there is a subset $I \subseteq [n]$ of cardinality m such that V_{int} is equal to the row space of $C[\![kn+I, *]\!]$. However, V_{int} is equal to the row space of $C[\![kn+I, *]\!]$ only if the support of $C[\![(j-1)n + \phi_j(i), *]\!]$ is equal to $\{(j-1)n^2 + (\phi_j(i)-1)n + i\}$ for every $(j, i) \in [k] \times I$. The later event occurs only if Bob can correctly guess $\phi_j(i)$ for every $(j, i) \in [k] \times I$. Since $\phi_1, \ldots, \phi_k$ are totally random, the last event happens with probability at most $\left(\frac{(n-m)!}{n!}\right)^k$. It follows that $\Pr[2 \leq \dim(V_{\text{int}}) \leq n] = \sum_{m=2}^{n} \Pr[\dim(V_{\text{int}}) = m] \leq \sum_{m=2}^{n} \binom{n}{m} \left(\frac{(n-m)!}{n!}\right)^k \leq 2^{-k}$, where the probabilities are taken over the random permutations $\phi_1, \ldots, \phi_k \leftarrow \mathbb{S}_n$. □

Let $H = \{t \in [kn] : C[\![t, *]\!] \neq \mathbf{0}\}$ and V_{int} be equal to the row space of $C[\![kn+I, *]\!]$, where $I \subseteq [n]$ is of cardinality m. Then we have the following lemma.

Lemma 6.1.7 *Let $\alpha \in \mathbb{F}^{|H|}$ and $\beta \in \mathbb{F}^m$ be such that $\Pr[Y[\![H]\!] = \alpha, s[\![I]\!] = \beta] > 0$. Then for any $\gamma \in \mathbb{F}^{n-m}$, $\Pr\left[s[\![\bar{I}]\!] = \gamma | Y[\![H]\!] = \alpha, s[\![I]\!] = \beta\right] = |\mathbb{F}|^{m-n}$, where the probabilities are taken over the random matrices $X_1, \ldots, X_k$ in equation* (6.1).

Proof: Clearly, $\dim(V_{\text{que}}) = |H|$, $\dim(V_{\text{sec}}) = n$ and $\dim(V_{\text{int}}) = m$. It follows that $\dim(V_{\text{que}} + V_{\text{sec}}) = \dim(V_{\text{que}}) + \dim(V_{\text{sec}}) - \dim(V_{\text{int}}) = |H| + n - m$. Consider the following linear equations of unknowns X: (i) $C[\![H, *]\!] \cdot X = \alpha$, (ii) $C[\![kn+I, *]\!] \cdot X = \beta$ and (iii) $C[\![kn + \bar{I}, *]\!] \cdot X = \gamma$. It is not hard to see that the equation systems (i)-(ii) and (i)-(ii)-(iii) have $|\mathbb{F}|^{kn^2-|H|}$ and $|\mathbb{F}|^{kn^2-|H|-n+m}$ solutions, respectively. Hence, $\Pr\left[s[\![\bar{I}]\!] = \gamma | Y[\![H]\!] = \alpha, s[\![I]\!] = \beta\right] = |\mathbb{F}|^{kn^2-|H|-n+m} \cdot |\mathbb{F}|^{-(kn^2-|H|)} = |\mathbb{F}|^{m-n}$, where the probability is taken over the random matrices $X_1, \ldots, X_k$ in equation (6.1). □

Lemma 6.1.7 intuitively shows that the receiver Bob learns essentially no information on the remaining secrets if he is able to obtain m out of the n secrets in an execution of π^g.

Lemma 6.1.8 *Let $R = H \bigcap [(k-1)n]$. Then for every $\alpha \in \mathbb{F}^{|R|}$ and $\beta \in \mathbb{F}^n$, $\Pr[Y[\![R]\!] = \alpha | s = \beta] = |\mathbb{F}|^{-|R|}$, where the probability is taken over the random matrices $X_1, \ldots, X_k$ in* (6.1).

Proof: The proof is similar to that of Lemma 6.1.7 and omitted. □

Lemma 6.1.8 shows that, among the first $(k-1)n$ answers Bob receives from $\mathcal{T}^g$, those indexed by R are always totally random. On the other hand, the answers indexed by $[(k-1)n] \setminus R$ are always 0.

Lemma 6.1.9 *For every $i \in I$, there is a vector $\rho \in \mathbb{F}^k$ such that*

$$Y_{(k-1)n+\phi_k(i)} = \rho_k^{-1} \cdot \Big(s_i - \sum_{j=1}^{k-1} \rho_j \cdot Y_{(j-1)n+\phi_j(i)} \Big). \tag{6.3}$$

Proof: Due to the proof of Lemma 6.1.5, we have that $C[\![kn+i, *]\!] \in \text{span}\big\{C[\![(\lambda-1)n+\tau, *]\!] : (\lambda,\tau) \in \Omega_i\big\}$, where $\Omega_i = \{(\lambda,\tau) : (\lambda-1)n+\tau \in P_i\}$. More precisely, by (6.2), we have that $C[\![kn+i, *]\!] \in \text{span}\big\{C[\![(j-1)n+\phi_j(i), *]\!] : j \in [k]\big\}$. It follows that there is a vector $\rho \in \mathbb{F}^k$ such that

$$C[\![kn+i, *]\!] = \sum_{j=1}^{k} \rho_j \cdot C[\![(j-1)n+\phi_j(i), *]\!]. \tag{6.4}$$

Due to equation (6.1), we multiply both sides of (6.4) by X and have that $s_i = \sum_{j=1}^{k} \rho_j \cdot Y_{(j-1)n+\phi_j(i)}$, which yields Equality (6.3). □

Lemma 6.1.9 shows that if a secret s_i can be obtained by Bob, then the last answer (i.e., $Y_{(k-1)n+\phi_k(i)}$) regarding to s_i is always uniquely determined by the first $k-1$ answers (i.e., $Y_{(j-1)n+\phi_k(i)}$, where $1 \le j \le k-1$) regarding to s_i. The result is actually not surprising since each secret s_i is additively shared among k random matrices $X_1, \ldots, X_k$ chosen by Alice and each answer regarding to s_i is uniquely determined by one of the matrices.

Lemma 6.1.10 *Let $i \in \bar{I}$ be such that $C[\![(k-1)n+\phi_k(i), *]\!] \neq \mathbf{0}$. Then for every $\alpha \in \mathbb{F}^{|R_i|}, \beta \in \mathbb{F}^n$ and $\gamma \in \mathbb{F}$, $\Pr[Y_{(k-1)n+\phi_k(i)} = \gamma | Y[\![R_i]\!] = \alpha, s = \beta] = |\mathbb{F}|^{-1}$, where $R_i = H \bigcap [(k-1)n+\phi_k(i)-1]$ and the probability is taken over the random matrices $X_1, \ldots, X_k$ in (6.1).*

Proof: The proof is similar to that of Lemma 6.1.7 and omitted. □

Lemma 6.1.10 shows that if $i \in \bar{I}$, then the receiver Bob essentially learns no information on s_i.

Lemma 6.1.11 *For any g-hybrid model adversary $\mathcal{H}$ corrupting Bob, there is an ideal model adversary $\mathcal{S}$ corrupting Bob whose running time is polynomial in that of $\mathcal{H}$ such that* $\mathrm{IDEAL}_{f,\mathcal{S}} \approx \mathrm{EXEC}_{\pi^g,\mathcal{H}}$.

Proof: The ideal model adversary $\mathcal{S}$ depicted by Figure 6-3 either fails or halt with *counter* $\in \{0,1\}$. It fails if and only if the event $\dim(V'_{\mathrm{int}}) \geq 2$ is detected at step 6.

The event $\dim(V'_{\mathrm{int}}) \geq 2$ indicates that Bob can obtain more than one secret. Due to Lemma 6.1.6, we have that

$$\Pr[\mathcal{S} \text{ fails}] = \Pr[\text{ the event } \dim(V'_{\mathrm{int}}) \geq 2 \text{ is detected at step 6}] \leq 2^{-k}.$$

Therefore, it is enough to show that

$$\mathrm{IDEAL}_{f,\mathcal{S}}(k,s,c,z) \equiv \mathrm{EXEC}_{\pi^g,\mathcal{H}}(k,s,c,z)$$

for every (k,s,c,z) which causes $\mathcal{S}$ to halt with *counter* $\in \{0,1\}$. For any such (k,s,c,z), let

$$\mathbb{V} = \big\{(c_{ji}, Y_{ji}) : (j,i) \in [k] \times [n]\big\}$$

be the set of queries and answers involving Bob in the g-hybrid model. Then we have

$$\mathrm{EXEC}_{\pi^g,\mathcal{H}}(k,s,c,z) = (\mathcal{H}_2(k,c,z,\mathbb{V},\phi), \perp, \perp),$$

where $\phi = (\phi_1, \ldots, \phi_k)$ are the random permutations chosen by Alice in the g-hybrid model. On the other hand, due to the construction of $\mathcal{S}$, we have that

$$\mathrm{IDEAL}_{f,\mathcal{S}}(k,s,c,z) = (\mathcal{H}_2(k,c,z,\mathbb{V}',\phi'), \perp, \perp).$$

We note that both ϕ and ϕ' consist of random permutations in $\mathbb{S}_n$. Hence, it suffices to show that the conditional distribution of $\mathbb{V}'$ given ϕ' and the conditional distribution

of $\mathbb{V}$ given ϕ are identical.

– input: (k, c, z), where $c \in [n]$ is Bob's input and z is an auxiliary input;
– subroutine: g-hybrid model adversary $\mathcal{H}$ and the trusted third party $\mathcal{T}^f$.

1. choose $\phi' = (\phi'_1, \ldots, \phi'_k) \leftarrow \mathbb{S}_n^k$ and define for every $i \in [n]$:

$$P'_i = \{(j-1)n + \phi'_j(i) : j \in [k]\}; Q'_i = \{(j-1)n^2 + (\phi'_j(i)-1)n + i : j \in [k]\}$$

2. initialize a $(kn+n) \times kn^2$ matrix C' s.t. :

$$C'[\![(j-1)n+i, *]\!] = \mathbf{0}; C'[\![kn+i, Q'_i]\!] = \mathbf{1}; C'[\![kn+i, \bar{Q}'_i]\!] = \mathbf{0};$$

3. initialize k all-zero square matrices $X'_1, \ldots, X'_k$ of order n;
4. let $T = [kn]$. initialize the following subspaces of the row space of C':

$$V'_{\text{que}} = \text{span}\{C'[\![T, *]\!]\}; V'_{\text{sec}} = \text{span}\{C'[\![\bar{T}, *]\!]\}; V'_{\text{int}} = V'_{\text{que}} \bigcap V'_{\text{sec}}.$$

5. initialize *counter* $= 0$. For $(j, i) = (1, 1), \ldots, (1, n), \ldots, (k, n)$,

- if $(j, i) \neq (1, 1)$, set $\mathbb{C}'_{ji} = \{(c'_{\lambda\tau}, Y'_{\lambda\tau}) : (\lambda, \tau) \in [k] \times [n], \lambda n + \tau < jn + i\}$;
- if $(j, i) = (1, 1)$, feed $\mathcal{H}$ with (k, c, z) and receive $c'_{ji} = \mathcal{H}_1(k, c, z)$;
- if $(j, i) \neq (1, 1)$, feed $\mathcal{H}$ with $(k, c, z, \mathbb{C}'_{ji})$ and receive $c'_{ji} = \mathcal{H}_1(k, c, z, \mathbb{C}'_{ji})$;
- update the $(j-1)n + i$th row of C' s.t. $C'[\![(j-1)n+i, S_{ji}]\!] = c'_{ji}$;
- update the vector spaces $V'_{\text{que}}, V'_{\text{sec}}, V'_{\text{int}}$ and check the value of *counter*,
  - if $\dim(V'_{\text{int}}) = 0$ or $\dim(V'_{\text{int}}) = 1$ and *counter* $= 1$, choose $x'_{ji} \leftarrow \mathbb{F}^n$, update the matrix X'_j s.t. $X'_j[\![i, *]\!] = x'_{ji}$ and set $Y'_{ji} = c'_{ji} \cdot x'_{ji}$;
  - if $\dim(V'_{\text{int}}) \geq 2$, output a failure message and halt;
  - if $\dim(V'_{\text{int}}) = 1$ and *counter* $= 0$ (we have that $j = k$)
    - find $c' \in [n]$ s.t. $V'_{\text{int}} = \text{span}\{C'[\![kn + c', *]\!]\}$;
    - find $\rho' \in \mathbb{F}^k$ s.t. $C'[\![kn+c', *]\!] = \sum_{j=1}^k \rho'_j \cdot C'[\![(j-1)n+\phi'_j(c'), *]\!]$;
    - feed $\mathcal{T}$ with c' and receive $s_{c'}$ from $\mathcal{T}$;
    - choose $x'_{ki} \leftarrow \mathbb{F}^n$ s.t. $s_{c'} = \sum_{j=1}^{k-1} \rho'_j \cdot Y'_{j\ \phi'_j(c')} + \rho'_k \cdot C'[\![(k-1)n + \phi'_k(c'), *]\!] \cdot x'_{ki}$;
    - update the matrix X'_k s.t. $X'_k[\![i, *]\!] = x'_{ki}$ and set $Y'_{ki} = c'_{ki} \cdot x'_{ki}$;
    - *counter* $=$ *counter* $+ 1$.

6. feed $\mathcal{H}$ with $(k, c, z, \mathbb{V}', \phi')$ and output whatever (say $\mathcal{H}_2(k, c, z, \mathbb{V}', \phi')$) $\mathcal{H}$ outputs, and then halt, where $\mathbb{V}' = \{(c'_{ji}, Y'_{ji}) : (j, i) \in [k] \times [n]\}$.

Figure 6-3: Ideal model adversary corrupting Bob for π^g

For every $(j,i) \in [k] \times [n]$, we define

$$\mathbb{C}_{ji} \;=\; \big\{(c_{\lambda\tau}, Y_{\lambda\tau}) : (\lambda,\tau) \in [k] \times [n] \text{ and } (\lambda,\tau) < (j,i)\big\}$$

to be the counterpart of $\mathbb{C}'_{ji}$ in the g-hybrid model. Furthermore, we define two random variables as follows:

$$\mathbb{Y}'_{ji} = c'_{ji} \circ \mathbb{C}'_{ji}, \qquad \mathbb{Y}_{ji} = c_{ji} \circ \mathbb{C}_{ji},$$

where $\circ$ denotes the concatenation operation. We stress that the notations $\mathbb{V}', \mathbb{C}'_{ji}, \mathbb{Y}'_{ji}$, $\phi', \mathbb{V}, \mathbb{C}_{ji}, \mathbb{Y}_{ji}$ and ϕ are always treated as random variables.

We want to take specific elements from the domains of the above random variables and write the conditional probability of $\mathbb{V}$ given ϕ explicitly. For this purpose, we suppose $\mathbb{V}^\star, \mathbb{C}^\star_{ji}, \mathbb{Y}^\star_{ji}$ and $\phi^\star$ are elements (i.e., not random variables) in the domains of $\mathbb{V}, \mathbb{C}_{ji}, \mathbb{Y}_{ji}$ and ϕ, respectively. It follows that

$$\begin{aligned}\Pr[\mathbb{V} = \mathbb{V}^\star | \phi = \phi^\star] = \prod_{(j,i)\in[k]\times[n]} & \Pr[c_{ji} = c^\star_{ji} \mid \mathbb{C}_{ji} = \mathbb{C}^\star_{ji}, \phi = \phi^\star] \\ & \cdot \Pr[Y_{ji} = Y^\star_{ji} \mid \mathbb{Y}_{ji} = \mathbb{Y}^\star_{ji}, \phi = \phi^\star].\end{aligned}$$

Clearly, this identity holds as well if we replace the $\mathbb{V}, \phi, c_{ji}, Y_{ji}$ with $\mathbb{V}', \phi', c'_{ji}, Y'_{ji}$, respectively. Therefore, it is sufficient to show that for every $(j,i) \in [k] \times [n]$,

(i) the conditional distributions of c'_{ji} and c_{ji} given $\mathbb{C}^\star_{ji}$ and $\phi^\star$ are identical;

(ii) the conditional distributions of Y'_{ji} and Y_{ji} given $\mathbb{Y}^\star_{ji}$ and $\phi^\star$ are identical.

We note that the space V'_{int} is uniquely determined by $\mathbb{V}'$ and ϕ' at step 5 of Adversary 2. We suppose that $V^\star_{\text{int}}$ is the space determined by $\mathbb{V}^\star$ and $\phi^\star$ in the same way. Since $\mathbb{V}^\star, \mathbb{C}^\star_{ji}, \mathbb{Y}^\star_{ji}$ and $\phi^\star$ are associated with the (k,s,c,z) which causes $\mathcal{S}$ to halt with *counter* $\in \{0,1\}$, either $\dim(V^\star_{\text{int}}) = 0$ or $\dim(V^\star_{\text{int}}) = 1$. Therefore, we have two cases for the proof of (i) and (ii):

Case I. $\dim(V^\star_{\text{int}}) = 0$.

For $(j,i) = (1,1)$, we have that $c'_{11} = \mathcal{H}_1(k,c,z)$ and $c_{11} = \mathcal{H}_1(k,c,z)$, which implies that the conditional distributions of c'_{11} and c_{11} given $\phi^\star$ are identical. Clearly, $\mathbb{Y}^\star_{11} = \{c^\star_{11}\}$. Given $\mathbb{Y}^\star_{11}$ and $\phi^\star$, we have that

$$\begin{aligned} Y'_{11} &= c'_{11} \cdot X'_1[\![1,*]\!] = c^\star_{11} \cdot X'_1[\![1,*]\!], \\ Y_{11} &= c_{11} \cdot X_1[\![1,*]\!] = c^\star_{11} \cdot X_1[\![1,*]\!], \end{aligned}$$

where $X'_1[\![1,*]\!], X_1[\![1,*]\!] \leftarrow \mathbb{F}^n$ are chosen by $\mathcal{S}$ and $\mathcal{H}$, respectively. Clearly, we have that $Y'_{11} \equiv Y_{11}$. It follows that (i) and (ii) hold for $(j,i) = (1,1)$.

Suppose that (i) and (ii) hold for every (j,i) such that $jn + i \leq \mu n + \nu$. Let $(\lambda,\tau) \in [k] \times [n]$ be such that $(\lambda-1)n + \tau = (\mu-1)n + \nu + 1$. We want to show that (i) and (ii) hold for $(j,i) = (\lambda,\tau)$ as well. Given $\mathbb{C}^\star_{\lambda\tau}$ and $\phi^\star$, we have $c'_{\lambda,\tau} = \mathcal{H}_1(k,c,z,\mathbb{C}^\star_{\lambda\tau})$ and $c_{\lambda,\tau} = \mathcal{H}_1(k,c,z,\mathbb{C}^\star_{\lambda\tau})$, which implies that (i) holds for $(j,i) = (\lambda,\tau)$. Given that $\mathbb{Y}^\star_{\lambda\tau} = c^\star_{\lambda\tau} \circ \mathbb{C}^\star_{\lambda\tau}$, we have that

$$Y'_{\lambda\tau} = c'_{\lambda\tau} \cdot X'_\lambda[\![\tau,*]\!] = c^\star_{\lambda\tau} \cdot X'_\lambda[\![\tau,*]\!], \tag{6.5}$$

$$Y_{\lambda\tau} = c_{\lambda\tau} \cdot X_\lambda[\![\tau,*]\!] = c^\star_{\lambda\tau} \cdot X_\lambda[\![\tau,*]\!]. \tag{6.6}$$

where $X'_\lambda[\![\tau,*]\!] \leftarrow \mathbb{F}^n$ is chosen by $\mathcal{S}$ at step 6 due to $\dim(V'_{\text{int}}) = \dim(V^\star_{\text{int}}) = 0$ and $X_\lambda[\![\tau,*]\!]$ is chosen by Alice in the g-hybrid model subject to the last n equations of (6.1). If $c^\star_{\lambda\tau} = \mathbf{0}$, then $Y'_{\lambda\tau} = Y_{\lambda\tau} = 0$. If $c^\star_{\lambda\tau} \neq \mathbf{0}$, then due to Lemma 6.1.8 and Lemma 6.1.10, the answer $Y_{\lambda\tau}$ is uniformly distributed. On the other hand, $Y'_{\lambda\tau}$ is uniformly distributed due to the choice of $X'_\lambda[\![\tau,*]\!]$. Therefore, (6.6) holds for $(j,i) = (\lambda,\tau)$.

Case II. $\dim(V^\star_{\text{int}}) = 1$.

Due to Lemma 6.1.5, there is an index $c^\star \in [n]$ such that $V^\star_{\text{int}} = \text{span}\big\{C^\star[\![kn + c^\star,*]\!]\big\}$ and $\rho^\star \in \mathbb{F}^k$ such that

$$C^\star[\![kn + c^\star,*]\!] = \sum_{j=1}^{k} \rho^\star_j \cdot C^\star[\![(j-1)n + \phi^\star_j(c^\star),*]\!]. \tag{6.7}$$

Let $(\lambda, \tau) = (k, \phi_k^\star(c^\star))$. Due to Lemma 6.1.8 and Lemma 6.1.10, it can be shown that (i) and (ii) hold for any (j, i) such that $jn + i < \lambda n + \tau$ and (i) also holds for $(j, i) = (\lambda, \tau)$ as we have done in CASE I. It remains to show that (ii) holds for $(j, i) = (\lambda, \tau)$. Given $\mathbb{Y}_{\lambda\tau}^\star$ and $\phi^\star$, we have (6.5) and (6.6) as well.

Due to (6.7) and Lemma 6.1.9, we have that $Y_{\lambda\tau} = (\rho_k^\star)^{-1}\Big(s_{c^\star} - \sum_{j=1}^{k-1} \rho_j^\star \cdot Y_{j\phi_j^\star(c^\star)}\Big)$. On the other hand, the ideal model adversary $\mathcal{S}$ should note that $\dim(V'_{\text{int}}) = 1$ and *counter* $= 0$ when $(j, i) = (\lambda, \tau)$ at step 6. Then it sets $c' = c^\star, \rho' = \rho^\star$ and decides $Y'_{\lambda\tau}$ subject to $\sum_{j=1}^{k} \rho'_j \cdot Y'_{j\phi'_j(c')} = s_{c'}$. Hence, we have that

$$Y'_{\lambda\tau} = (\rho_k^\star)^{-1}\left(s_{c^\star} - \sum_{j=1}^{k-1} \rho_j^\star \cdot Y_{j\phi_j^\star(c^\star)}\right),$$

which is equal to $Y_{\lambda\tau}$. Therefore, (ii) holds for $(j, i) = (\lambda, \tau)$.

For every (j, i) such that $jn + i > \lambda n + \tau$, it is straightforward to show that the conditional distributions of c'_{ji} and c_{ji} given $\mathbb{C}_{ji}^\star$ and $\phi^\star$ are identical. Due to Lemma 6.1.10, the conditional distribution of Y'_{ji} and Y_{ji} given $\mathbb{Y}_{ji}^\star$ and $\phi^\star$ are both uniform if $c_{ji}^\star \neq \mathbf{0}$. Therefore, (i) and (ii) hold when $jn + i > \lambda n + \tau$.

Combining CASE I and CASE II, the conditional distribution of $\mathbb{V}'$ given ϕ' and the conditional distribution of $\mathbb{V}$ given ϕ are identical to each other.

□

By Lemma 6.1.1, 6.1.2 and 6.1.11, we have the following theorem:

Theorem 6.1.1 *There is a statistically secure reduction from* $\binom{n}{1}$-OT *to* $\mathbb{F}^n$-OLFE$_n$.

Let $\mathcal{C} \subseteq \mathbb{F}^n$ contain all unit vectors in $\mathbb{F}^n$ and $g' = \mathcal{C}$-OLFE$_n$. Let $\pi^{g'}$ be obtained from π^g by substituting g with g'. By the proof of Lemma 6.1.6, any hybrid model adversary $\mathcal{H}$ corrupting Bob in $\pi^{g'}$ cannot succeed with a larger probability than it does in π^g because $\mathcal{C} \subseteq \mathbb{F}^n$. Hence, we have that

Corollary 6.1.1 *There is a statistically secure reduction from* $\binom{n}{1}$-OT *to* $\mathcal{C}$-OLFE$_n$, *where* $\mathcal{C} \subseteq \mathbb{F}^n$ *contains all unit vectors in* $\mathbb{F}^n$.

6.2 OLFE and the Reversibility of OT

In this section, we present a $\mathcal{C}$-OLFE$_n$ which can be applied to reverse any $\binom{n}{1}$-OT, where $\mathcal{C} = \{(c_1, \ldots, c_n) : c_1 \oplus \cdots \oplus c_n = 1\} \subseteq \mathbb{F}_2^n$. More precisely, we present a $\mathcal{C}$-OLFE$_n$ from Alice(as a sender) to Bob(as a receiver) by reducing it, in a perfectly secure way, to $n-1$ invocations of a given $\binom{n}{1}$-OT from Bob(as a sender) to Alice (as a receiver). Let g be the $\mathcal{C}$-OLFE$_n$ and h be the $\binom{n}{1}$-OT. Figure 6-4 is a two-party protocol σ^h for g in the h-hybrid model.

- input: Alice has n bits $b \in \{0,1\}^n$ and Bob has a choice vector $c \in \mathcal{C}$;
- subroutine: the trusted third party $\mathcal{T}^h$;

1. Bob: choose $r_i \leftarrow \{0,1\}$ and set $a_{ij} = r_i \oplus (j-1) \cdot c_i$ for $2 \le i \le n$ and $j \in [n]$;
2. $\mathcal{T}^h$: for $i = 2 \ldots n$, Bob and Alice proceed as follows:
   - Bob: send $(a_{i1}, \cdots, a_{in})$ to $\mathcal{T}^h$;
   - Alice: send $d_i = b_1 \oplus b_i$ to $\mathcal{T}^h$ and receive x_i from $\mathcal{T}^h$;
3. Alice: send $y = b_1 \oplus x_2 \oplus \cdots \oplus x_n$ to $\mathcal{R}$;
4. Bob: output $y \oplus r_2 \oplus \cdots \oplus r_n$.

Figure 6-4: A construction of $\mathcal{C}$-OLFE$_n$ out of $\binom{n}{1}$-OT (σ^h)

Lemma 6.2.1 *If Alice and Bob are honest, then* IDEAL$_{f,\mathcal{S}} \equiv$ EXEC$_{\sigma^h,\mathcal{H}}$, *where* $\mathcal{S}$ *is the ideal model adversary corrupting no party and* $\mathcal{H}$ *is the* h*-hybrid model adversary corrupting no party.*

Proof: Given (k, b, c, z), we have IDEAL$_{f,\mathcal{S}}(k, b, c, z) = (\perp, \perp, \bigoplus_{i=1}^n (b_i \cdot c_i)) = (\perp, \perp, b_1 \cdot (1 \oplus \bigoplus_{i=2}^n c_i) \oplus \bigoplus_{i=2}^n (b_i \cdot c_i)) = (\perp, \perp, b_1 \oplus \bigoplus_{i=2}^n (b_1 \oplus b_i) \cdot c_i) = (\perp, \perp, b_1 \oplus \bigoplus_{i=2}^n (r_i \oplus (b_1 \oplus b_i) \cdot c_i) \oplus \bigoplus_{i=2}^n r_i) = (\perp, \perp, y \oplus \bigoplus_{i=2}^n r_i) =$ EXEC$_{\sigma^h,\mathcal{H}}(k, b, c, z)$ where the random variables only depend on the uniform and independent coin tosses of Bob. $\square$

Lemma 6.2.2 *For any h-hybrid model adversary $\mathcal{H}$ corrupting Alice, there is an ideal model adversary $\mathcal{S}$ corrupting Alice whose running time is polynomial in that of $\mathcal{H}$ such that* $\text{IDEAL}_{f,\mathcal{S}} \equiv \text{EXEC}_{\sigma^h,\mathcal{H}}$.

- input: (k, b, z), where b is Alice's input and z is an auxiliary input;
- subroutine: the h-hybrid model adversary $\mathcal{H}$ and the trusted third party $\mathcal{T}^g$;

1. feed $\mathcal{H}$ with (k, b, z);
2. for $i = 2, \ldots, n$ execute
   (a) receive d'_i from $\mathcal{H}$;
   (b) choose $x'_i \leftarrow \{0, 1\}$ and feed $\mathcal{H}$ with x'_i;
3. receive y' from $\mathcal{H}$;
4. set $b'_1 = y' \oplus x'_2 \oplus \ldots \oplus x'_n$ and $b'_i = b'_1 \oplus d'_i$ for every $i = 2, \ldots, n$;
5. feed $\mathcal{T}$ with $b'_1, \ldots, b'_n$;
6. output whatever $\mathcal{H}$ outputs, say $\mathcal{H}(k, b, z, d', x', y')$, where $d' = (d'_2, \ldots, d'_n)$ and $x' = (x'_2, \ldots, x'_n)$

Figure 6-5: Ideal model adversary corrupting Alice for σ^h

Proof: The ideal model adversary is depicted by Figure 6-5. Given (k, b, c, z), we have $\text{IDEAL}_{f,\mathcal{S}}(k, b, c, z) = (\mathcal{H}(k, b, z, d', x', y'), \perp, b' \cdot c) = (\mathcal{H}(k, b, z, d', x', y'), \perp, b'_1 \cdot (1 \oplus \bigoplus_{i=2}^n c_i) \oplus \bigoplus_{i=2}^n b'_i \cdot c_i) = (\mathcal{H}(k, b, z, d', x', y'), \perp, b'_1 \oplus \bigoplus_{i=2}^n (b'_1 \oplus b'_i) \cdot c_i) = (\mathcal{H}(k, b, z, d', x', y'), \perp, y' \oplus \bigoplus_{i=2}^n x'_i \oplus \bigoplus_{i=2}^n d'_i \cdot c_i) \equiv (\mathcal{H}(k, b, z, d, x, y), \perp, y \oplus \bigoplus_{i=2}^n x_i \oplus \bigoplus_{i=2}^n d_i \cdot c_i) = (\mathcal{H}(k, b, z, d, x, y), \perp, y \oplus \bigoplus_{i=2}^n (r_i \oplus d_i \cdot c_i) \oplus \bigoplus_{i=2}^n d_i \cdot c_i) = (\mathcal{H}(k, b, z, d, x, y), \perp, y \oplus \bigoplus_{i=2}^n r_i) = \text{EXEC}_{\sigma^h,\mathcal{H}}(k, b, c, z)$, where $b' = (b'_1, \ldots, b'_n), d = (d_2, \ldots, d_n)$ and $x = (x_2, \ldots, x_n)$. □

Lemma 6.2.3 *For any h-hybrid model adversary $\mathcal{H}$ corrupting Bob, there is an ideal model adversary $\mathcal{S}$ corrupting Bob whose running time is polynomial in that of $\mathcal{H}$ such that* $\text{IDEAL}_{f,\mathcal{S}} \equiv \text{EXEC}_{\sigma^h,\mathcal{H}}$.

– input: (k, c, z), where $c \in \mathcal{C}$ is Bob's input and z is an auxiliary input;

– subroutine: the h-hybrid model adversary $\mathcal{H}$ and the trusted third party $\mathcal{T}^g$.

1. feed $\mathcal{H}$ with (k, c, z) and receive $\left\{a'_{ij} : i = 2, \ldots, n \text{ and } j \in [n]\right\}$ from $\mathcal{H}$;
2. set $c'_i = a'_{i1} \oplus a'_{i2}$ for $i = 2, \ldots, n$ and $c'_1 = 1 \oplus c'_2 \oplus \ldots \oplus c'_n$;
3. feed $\mathcal{T}$ with $c' = (c'_1, \ldots, c'_n)$ and receive $\bigoplus_{i=1}^{n} b_i \cdot c'_i$;
4. feed $\mathcal{H}$ with $y' = \bigoplus_{i=2}^{n} a'_{i1} \oplus \bigoplus_{i=1}^{n} b_i \cdot c'_i$ and output whatever $\mathcal{H}$ outputs, say $\mathcal{H}(k, c, z, a', y')$, where $a' = \left\{a'_{ij} : i = 2, \ldots, n \text{ and } j \in [n]\right\}$.

Figure 6-6: Ideal model adversary corrupting Bob for σ^h

Proof: The ideal model adversary $\mathcal{S}$ is depicted by Figure 6-6. Given (k, b, c, z) and $a = \{a_{ij} : i = 2, \ldots, n \text{ and } j \in [n]\}$. We have that $\text{IDEAL}_{f,\mathcal{S}}(k, b, c, z) = (\mathcal{H}(k, c, z, a', y'), \perp, \perp)$ and $\text{EXEC}_{\sigma^h,\mathcal{H}}(k, b, c, z) = (\mathcal{H}(k, c, z, a, y), \perp, \perp)$. Clearly, $(a', y') = (a', \bigoplus_{i=2}^{n} a'_{i1} \oplus \bigoplus_{i=1}^{n} b_i \cdot c'_i) = (a', \bigoplus_{i=2}^{n} a'_{i1} \oplus (b_1 \cdot c'_1) \oplus \bigoplus_{i=2}^{n} b_i \cdot (a'_{i1} \oplus a'_{i2})) = (a', \bigoplus_{i=2}^{n} a'_{i1} \oplus b_1 \cdot (1 \oplus \bigoplus_{i=2}^{n} (a'_{i1} \oplus a'_{i2})) \oplus \bigoplus_{i=2}^{n} b_i \cdot (a'_{i1} \oplus a'_{i2})) \equiv (a, \bigoplus_{i=2}^{n} a_{i1} \oplus (b_1 \cdot (1 \oplus \bigoplus_{i=2}^{n} (a_{i1} \oplus a_{i2}))) \oplus \bigoplus_{i=2}^{n} b_i \cdot (a_{i1} \oplus a_{i2})) = (a, b_1 \oplus \bigoplus_{i=2}^{n} (a_{i1} \oplus (b_1 \oplus b_i) \cdot (a_{i1} \oplus a_{i2}))) = (a, y)$, which completes the proof of the lemma. □

Theorem 6.2.1 *There is a perfectly secure reduction from* $\mathcal{C}$-OLFE_n *to* $\binom{n}{1}$-OT, *where the choice space of the receiver is* $\mathcal{C} = \{(c_1, \ldots, c_n) : c_1 \oplus \ldots \oplus c_n = 1\} \subseteq \mathbb{F}_2^n$.

Note that $\mathcal{C}$ contains all unit vectors in $\mathbb{F}^n$. Therefore, by Corollary 6.1.1, the resulting $\mathcal{C}$-OLFE_n over $\mathbb{F}_2$ can be transformed to an $\binom{n}{1}$-OT except for a negligible failure probability. By the composition theorem of secure two-party protocols [19], we have the following theorem.

Theorem 6.2.2 *The* $\binom{n}{1}$-OT *from Alice to Bob can be statistically securely reduced to* $kn(n-1)$ *invocations of the* $\binom{n}{1}$-OT *from Bob to Alice.*

Theorem 6.2.2 shows that $\binom{n}{1}$-OT can be efficiently reversed to for *any* $n \geq 2$.

Chapter 7

Conclusion

In this thesis, we have studied four notions related to private information retrieval, namely extended private information retrieval, locally decodable codes, distributed oblivious transfer and oblivious linear function evaluation.

We showed that the Bringer-Chabanne EPIR protocol for polynomial evaluation fails frequently for a class of polynomials. It is an interesting open problem to correct their protocol or propose new EPIR protocols which meet the correctness requirement.

We showed that Mersenne numbers which are products of two primes give considerable savings within Efremenko's framework. Using fifty such numbers, we constructed new efficient LDCs and PIR protocols. Such numbers are said to be algebraically nice. It is an open problem to identify more algebraically nice (Mersenne) numbers.

We constructed communication-efficient DOT protocols from information-theoretic PIR protocols. These are the first DOT protocols of sublinear communication complexity. However, the setup complexity of our protocols are quite large. Therefore, our constructions are efficient for a limited range of parameters. It is an open problem to reduce the setup complexity of our constructions.

We introduced a new cryptographic primitive called $\mathcal{C}$-OLFE$_n$. It turns out to be a useful tool in cryptographic study and protocol design. Specifically, we show that it can be used to reverse the direction of any oblivious transfer. It is an interesting problem to find more applications of this primitive in cryptographic study and protocol design.

Bibliography

[1] Andris Ambainis. Upper bound on the communication complexity of private information retrieval. In Pierpaolo Degano, Roberto Gorrieri, Alberto Marchetti-Spaccamela (eds.) ICALP 1997. LNCS, vol. 1256, pp. 401-407. Springer, Heidelberg, 1997.

[2] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. In APPROX-RANDOM 2007. LNCS, vol. 4627, pp. 311-325. Springer, Heidelberg, 2007.

[3] Donald Beaver. Precomputing oblivious transfer. In Don Coppersmith (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 97-109. Springer, Heidelberg, 1995.

[4] Amos Beimel, Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. Communication-efficient distributed oblivious transfer. In Journal of Computer and System Sciences. To Appear, 2012.

[5] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. In Journal of Computer and System Sciences, vol. 71: 2, pp. 213-247, 2005.

[6] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. One-way functions are essential for single-server private information retrieval. In STOC 1999, pp. 89-98. ACM, New York, 1999.

[7] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-Francois Raymond. Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In FOCS 2002, pp. 261-270. IEEE, Los Alamitos, 2002.

[8] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 547-557. Springer, Heidelberg, 1990.

[9] C. H. Bennett, Gilles Brassard, Claude Crépeau, and H Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 351-366. Springer, Heidelberg, 1992.

[10] G. R. Blakley. Safeguarding cryptographic keys. In R. E. Merwin, J. T. Zanca, and M. Smith (eds.) Proceedings of the 1979 AFIPS National Computer Conference, vol. 48 of AFIPS Conference proceedings, pp. 313-317. AFIPS Press, 1979.

[11] Carlo Blundo, Paolo D'Arco, Alfredo De Santis, and Douglas R. Stinson. On unconditionally secure distributed oblivious transfer. In Journal of Cryptology, vol. 20, pp. 323-373, 2007.

[12] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. Information theoretic reductions among disclosure problems. In FOCS 1986, pp. 168-173. IEEE, Los Alamitos, 1986.

[13] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234-238. Springer, Heidelberg, 1987.

[14] Gilles Brassard, Claude Crépeau, and Stefan Wolf. Oblivious transfers and privacy amplification. In Journal of Cryptology, vol. 16: 4, pp. 219-237, 2003.

[15] Julien Bringer and Hervé Chabanne. Another look at extended private information retrieval protocols. In Bart Preneel (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 305-322. Springer, Heidelberg, 2009.

[16] Julien Bringer, Hervé Chabanne, David Pointcheval, and Qiang Tang. Extended private information retrieval and its application in biometrics authentications. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, Chaoping Xing (eds.) CANS 2007. LNCS, vol. 6467, pp. 175-193. Springer, Heidelberg, 2007.

[17] Christian Cachin. On the foundations of oblivious transfer. In Kaisa Nyberg (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 361-374. Springer, Heidelberg, 1998.

[18] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402-414. Springer, Heidelberg, 1999.

[19] Ran Canetti. Security and composition of multiparty cryptographic protocols. In Journal of Cryptology, vol. 13: 1, pp. 143-202, 2000.

[20] Ran Canetti, Yuval Ishai, Ravi Kumar, Michael K. Reiter, Ronitt Rubinfeld, and Rebecca N. Wright. Selective private function evaluation with applications to private statistics. In PODC 2001, pp. 293-304. ACM, New York, 2001.

[21] Yan-Cheng Chang. Single database private information retrieval with logarithmic communication. In Huaxiong Wang, Josef Pieprzyk, Vijay Varadharajan (eds.) ACISP 2004. LNCS, vol. 3108, pp. 50-61. Springer, Heidelberg, 2004.

[22] Yeow Meng Chee, Tao Feng, San Ling, Huaxiong Wang, and Liang Feng Zhang. Query-efficient locally decodable codes of subexponential length. In Computational Complexity. To Appear, 2011.

[23] Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. On Bringer-Chabanne EPIR protocol for polynomial evaluation. In Journal of Mathematical Cryptology. To Appear, 2011.

[24] Yeow Meng Chee, Huaxiong Wang, and Liang Feng Zhang. Oblivious transfer and n-variate linear function evaluation. In Bin Fu, Ding-Zhu Du (eds.) COCOON 2011. LNCS, vol. 6842, pp. 627-637. Springer, Heidelberg, 2011.

[25] K. Y. Cheong, Takeshi Koshiba, and Shohei Nishiyama. Strengthening the security of distributed oblivious transfer. In Colin Boyd, Juan Manuel González Nieto (eds.) ACISP 2009, vol. 5594, pp. 377-388. Springer, Heidelberg, 2009.

[26] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In FOCS 1995, pp. 41-50. IEEE, Los Alamitos, 1995.

[27] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In Journal of the ACM, vol. 45: 6, pp. 965-982, 1998.

[28] Claude Crépeau and Miklos Santha. On the reversibility of oblivious transfer. In Donald W. Davies (ed.) EUROCRYPT 1991, LNCS, vol. 547, pp. 106-113. Springer, Heidelberg, 1991.

[29] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In Don Coppersmith (ed.) CRYPTO 1995, vol. 963, pp. 110-123. Springer, Heidelberg, 1995.

[30] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Universal service-providers for database private information retrieval. In PODC 1998, pp. 91-100, 1998.

[31] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In Bart Preneel (ed.) EUROCRYPT 2000, vol. 1807, pp. 122-138. Springer, Heidelberg, 2000.

[32] Charles W. Curtis and Irving Reiner. Representation theory of finite groups and associative algebras. In AMS Chelsea Publishing, Providence, RI, xiv+689, 2006.

[33] Amit Deshpande, Rahul Jain, Telikepalli Kavitha, Jaikumar Radhakrishnan, and Satyanarayana V. Lokam. Better lower bounds for locally decodable codes. In CCC 2002, pp. 184-193. IEEE Computer Society, Washington, DC, USA, 2002.

[34] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In FOCS 2010, pp. 705-714. IEEE, Los Alamitos, 2010.

[35] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In STOC 2005, pp. 592-601. ACM, New York, 2005.

[36] Klim Efremenko. 3-query locally decodable codes of subexponential length. In STOC 2009, pp. 39-44. ACM, New York, 2009.

[37] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In Communications of the ACM, vol. 28: 6, pp. 637-647, 1985.

[38] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian (ed.) TCC 2005. LNCS, vol. 3378, pp. 303-324. Springer, Heidelberg, 2005.

[39] William I. Gasarch. A survey on private information retrieval. In Bulltin of European Association for Theoretical Computer Science, vol. 82, pp. 72-107, 2004.

[40] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In Lus Caires, Giuseppe F. Italiano, Lus Monteiro, Catuscia Palamidessi, Moti Yung (eds.) ICALP 2005. LNCS, vol. 3580, pp. 803-815. Springer, Heidelberg, 2005.

[41] Yael Gertner, S. Goldwasser, and Tal Malkin. A random server model for private information retrieval. In RANDOM 1998. LNCS, vol. 1518, pp. 200-217, 1998.

[42] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In STOC 1998, pp. 151-160. ACM, New York, 1998.

[43] Yael Gertner, S. Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In FOCS 2000, pp. 325-335. IEEE, Los Alamitos, 2000.

[44] Oded Goldreich. Foundations of Cryptography: Basic Applications. Cambridge University Press, Cambridge. 2004.

[45] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In Computational Complexity, vol. 15: 3, pp. 263-296, 2006.

[46] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In STOC 1987, pp. 218-229. ACM, New York, 1987.

[47] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In STOC 1985, pp. 291-304. ACM, New York, 1985.

[48] Parikshit Gopalan. A note on Efremenko's locally decodable codes. In Electronic Colloquium on Computational Complexity (ECCC) TR09-069, 2009.

[49] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. In Combinatorica, vol. 20: 1, pp. 71-85, 2000.

[50] Jens Groth, Aggelos Kiayias, and Helger Lipmaa. Multi-query computationally-private information retrieval with constant communication rate. In Phong Q. Nguyen, David Pointcheval (eds.) PKC 2010. LNCS, vol. 6056, pp. 107-123. Springer, Heidelberg, 2010.

[51] Iftach Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In Moni Naor (ed.) TCC 2004, vol. 2951, pp. 394-409. Springer, Heidelberg, 2004.

[52] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In STOC 1989, pp. 44-61. ACM, New York, 1989.

[53] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Sufficient conditions for collision-resistant hashing. In Joe Kilian (ed.) TCC 2005, vol. 3378, pp. 445-456. Springer, Heidelberg, 2005.

[54] Mitsuru Ito, Akira Satio, and Takao Nishizeki. Secret sharing scheme realizing general access structure. In IEEE Global Telecommunications Conference, 1987.

[55] Toshiya Itoh and Yasuhiro Suzuki. New constructions for query-efficient locally decodable codes of subexponential length. In IEICE Transactions on Information and Systems, vol. E93-D: 2, pp. 263–270, 2010.

[56] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer (ed.) EUROCRYPT 2005, vol. 3494, pp. 78-95. Springer, Heidelberg, 2005.

[57] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In STOC 2000, pp. 80-86. ACM, New York, 2000.

[58] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of Mersenne numbers. In SIAM Journal on Computing, vol. 38: 5, pp. 1952-1969, 2009.

[59] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In Journal of Computer and System Science, vol. 69: 3, pp. 395-420, 2004.

[60] Joe Kilian. Founding cryptography on oblivious transfer. In STOC 1988, pp. 20-31. ACM, New York, 1988.

[61] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In STOC 2011, pp. 167-176. ACM, New York, 2011.

[62] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In FOCS 1997, pp. 364-373. IEEE, Los Alamitos, 1997.

[63] Eyal Kushilevitz and Rafail Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In Bart Preneel (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 104-121. Springer, Heidelberg, 2000.

[64] Leonid Levin László Babai, Lance Fortnow and Mario Szegedy. Checking computations in polylogarithmic time. In STOC 1991, pp. 21-31. ACM, New York, 1991.

[65] Rudolf Lidl and Harald Niederreiter. Finite Fields. In Second edition, Cambridge University Press, 1997.

[66] F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correcting Codes. In North-Holland Publishing Company, Amsterdam, 1977.

[67] Bernard R. McDonald. Finite rings with identity. In Marcel Dekker Inc, New York, ix+429. Pure and Applied Mathematics, vol. 28, 1974.

[68] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In STOC 1999, pp. 245-354. ACM, New York, 1999.

[69] Moni Naor and Benny Pinkas. Distributed oblivious transfer. In ASIACRYPT 2000. LNCS, vol. 1976, pp. 205-219. Springer, Heidelberg, 2000.

[70] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In SODA 2001, pp. 448-457. ACM, New York, 2001.

[71] Moni Naor and Benny Pinkas. Oblivious polynomial evaluation. In SIAM Journal on Computing, vol. 35: 5, pp. 1254-1281, 2006.

[72] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In ACM Conference on Electronic Commerce 1999, pp 129-139, 1999.

[73] Ventzislav Nikov, Svetla Nikova, Bart Preneel, and Joos Vandewalle. On unconditionally secure distributed oblivious transfer. In INDOCRYPT 2002. LNCS, vol. 2551, pp. 395-408. Springer, Heidelberg, 2002.

[74] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In Randomization and Approximation Techniques in Computer Science, LNCS, vol. 2483, pp. 39-50. Springer, Berlin, 2002.

[75] Rafail Ostrovsky and Victor Shoup. Private information storage. In STOC 1997, pp. 294-303. ACM, New York, 1997.

[76] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Fair games against an all-powerful adversary. In DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 13, pp. 155-169, 1990.

[77] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner (ed.) CRYPTO 2008, vol. 5157, pp. 554-571. Springer, Heidelberg, 2008.

[78] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In STOC 2008, pp. 187-196. ACM, New York, 2008.

[79] Alexander Polishchuk and Daniel Spielman. Nearly-linear size holographic proofs. In STOC 1994, pp. 194-203. ACM, New York, 1994.

[80] Michael O. Rabin. How to exchange secrets by oblivious transfer. In Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[81] Prasad Raghavendra. A note on yekhanin's locally decodable codes. In Electronic Colloquium on Computational Complexity (ECCC) TR07-016, 2007.

[82] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. In IEEE Transactions on Information Theory, vol. 4, pp. 38-49, 1954.

[83] Ronald Rivest. Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer. unpublished manuscript. In http://theory.lcs.mit.edu/ rivest/publications.html, 1999.

[84] Adi Shamir. How to share a secret. In Communications of the ACM, vol. 22, pp. 612-613, 1979.

[85] Dungjade Shiowattana and Satyanarayana V. Lokam. An optimal lower bound for 2-query locally decodable linear codes. In Information Processing Letters, vol. 97: 6, pp. 244-250, 2006.

[86] Madhu Sudan. Efficient checking of polynomials and proofs and the hardness of approximation problems. In PhD thesis, University of California at Berkeley, 1992.

[87] Luca Trevisan. Some applications of coding theory in computational complexity. In Electronic Colloquium on Computational Complexity (ECCC) TR04-043, 2004.

[88] Lawrence C. Washington. Introduction to cyclotomic fields. In volume 83 of Graduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition, xiv+487, 1997.

[89] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In ICALP 2005, LNCS, vol. 3580, pp. 1424-1436. Springer, Berlin, 2005.

[90] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay (ed.) EUROCRYPT 2006, LNCS, vol. 4004, pp. 222-232. Springer, Heidelberg, 2006.

[91] David P. Woodruff. New lower bounds for general locally decodable codes. In Electronic Colloquium on Computational Complexity (ECCC) TR07-006, 2007.

[92] David P. Woodruff and Sergey Yekhanin. A geometric approach to information-theoretic private information retrieval. In CCC 2005, pp. 275-284. IEEE Los Alamos, 2005.

[93] Akihiro Yamamura and Taiichi Saito. Private information retrieval based on the subgroup membership problem. In Vijay Varadharajan, Yi Mu (eds.) ACISP 2001. LNCS, vol. 2119, pp. 206-220. Springer, Heidelberg, 2001.

[94] Andrew Chi-Chih Yao. Protocols for secure computations. In FOCS 1982, pp. 160-164. IEEE, Los Alamitos, 1982.

[95] Andrew Chi-Chih Yao. How to generate and exchange secrets. In FOCS 1986, pp. 162-167. IEEE, Los Alamitos, 1986.

[96] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. In STOC 2007, pp. 266-274. ACM, New York, 2007.

[97] Sergey Yekhanin. Locally decodable codes: A brief survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao,Yuansheng Tang, Huaxiong Wang, Chaoping Xing (eds.) IWCC 2011. LNCS, vol. 6639, pp. 273-282. Springer, Heidelberg, 2011.