EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

Compact lossy and all-but-one trapdoor functions from lattice ★

Leixiao Cheng[1], Quanshui Wu[1], Yunlei Zhao[2,*]

[1]School of Mathematical Sciences, Fudan University, Shanghai (200433), China
[2]School of Computer Science, Fudan University, Shanghai (201203), China

Abstract

Lossy trapdoor functions (LTDF) and all-but-one trapdoor functions (ABO-TDF) are fundamental cryptographic primitives. And given the recent advances in quantum computing, it would be much desirable to develop new and improved lattice-based LTDF and ABO-TDF. In this work, we provide more compact constructions of LTDF and ABO-TDF based on the learning with errors (LWE) problem. In addition, our LWE-based ABO-TDF can allow smaller system parameters to support super-polynomially many injective branches in the construction of CCA secure public key encryption. As a core building tool, we provide a more compact homomorphic symmetric encryption schemes based on LWE, which might be of independent interest. To further optimize the ABO-TDF construction, we employ the full rank difference encoding technique. As a consequence, the results presented in this work can substantially improve the performance of all the previous LWE-based cryptographic constructions based upon LTDF and ABO-TDF.

Keywords: All-but-one trapdoor functions, Homomorphic symmetric encryption, Lattice, Learning with errors, Lossy trapdoor functions

Received on 21 December 2017; accepted on 26 December 2017; published on 28 December 2017

doi:10.4108/eai.28-12-2017.153517

1. Introduction

It is well known that trapdoor functions (TDFs) and security under chosen ciphertext attack (CCA security)[1–3] are very important notions in public-key cryptosystem.

Injective one-way trapdoor function F specifies, for each public key pk, a *deterministic* map F_{pk} that can be inverted given an associated trapdoor. It was one of the first abstract cryptographic primitives, allowing us to go back to the seminal paper of Diffie and Hellman [4]. TDFs had been constructed only from problems related to factoring [5–7] prior to the seminal work [8].

Adaptive chosen ciphertext attack deals with active attacks. Given an encryption of the target message, we want to guarantee that the adversary cannot obtain any partial information about the message, even in the presence of "decryption oracle". Obliviously, the adversary is not allowed to submit the target ciphertext itself to the oracle. If the adversary has access to the decryption oracle only *prior* to obtaining the target ciphertext, and the goal of the adversary is to obtain partial information about the encrypted message, then this type of attack is called a chosen ciphertext attack. CCA-secure cryptosystems had been realized based on problems related to factoring and discrete logs [1, 3, 9–11] using NIZK proofs but not lattices prior to the work [8].

The notion of lossy trapdoor functions (LTDF) was proposed by Peikert and Waters at STOC 2008 [8], which can be viewed as a strictly stronger powerful primitive than TDF. Informally speaking, a family of lossy trapdoor functions contains two computationally indistinguishable types of functions: injective functions with a trapdoor, and lossy functions that statistically lose information about their input. Furthermore, Lossiness implies one-wayness [8]. They imply many cryptographic primitives such as one-way trapdoor function [4], collision resistant hash function [12], oblivious transfer protocol [13], chosen ciphertext secure public key encryption scheme [1, 3, 8, 14], deterministic public key encryption scheme [15], OAEP based public key encryption scheme [16], and selective opening secure public key encryption

★This research was supported in part by NSFC (Grant Nos. 61472084 and U1536205), National Key R&D Program of China (No.2017YFB0802000), Shanghai innovation action project No. 16DZ1100200, and Shanghai science and technology development funds No.16JC1400801.
*Corresponding author. Email: ylzhao@fudan.edu.cn

scheme [17]. LTDF can be constructed based on many assumptions [8, 16, 18–20] and, in particular, lattice-based assumption (specifically, the LWE assumption) [8]. The learning with errors (LWE) problem was defined by Regev [21], which is a generalization of the well-known learning parity with noise problem to moduli larger than 2 [22]. Regev [21] showed that LWE is hard if the standard lattice problems are hard in the worst case for quantum algorithms. In fact, lattice-based constructions are especially desirable in the post-quantum era, since lattice-based cryptosystems are commonly believed to be resistant to quantum attacks.

In order to construct CCA-secure cryptosystem from LTDF, it is more convenient to consider a new notion called *all-but-one* trapdoor functions (ABO-TDF) [8]. In an ABO-TDF collection, each function has a lot of branches: only a single branch is lossy, while super-polynomially many branches are injective trapdoor functions owning the same trapdoor. In the construction of CCA-secure cryptosystems from ABO-TDF [8], an injective branch of ABO-TDF corresponds to the verification key of a one-time signature that is, in turn, used in forming the ciphertext. As a consequence, we expect to allow smaller parameters to support enough branches since super-polynomially many branches are needed in construction of CCA secure PKE. The basic relation between the two notions is revealed in [8]: lossy and ABO trapdoor functions are equivalent on appropriately chosen parameters. In this work, following the general paradigm proposed in [8] we provide improved and more compact constructions of LTDF and ABO-TDF based on the LWE problem. As a core building tool we provide a more compact homomorphic symmetric encryption schemes based on LWE, which might be of independent interest. To further reduce the size of the encrypted matrix of function indices of ABO-TDF, we make use of the full rank difference encoding (FRD) proposed in [23] (instead of the pairwise independent hash function originally used in [8]); The FRD technique not only reduces the matrix size, but also can allow smaller system parameters to support super-polynomially many injective branches in the construction of CCA secure public key encryption, which further optimize the construction of ABO-TDF. As a consequence, the results presented in this work can substantially improve the performance of all the previous LWE-based cryptographic constructions based upon LTDF and ABO-TDF.

2. Preliminaries

For a vector $\mathbf{x}$, $\mathbf{x}[i]$ denotes its i-th coordinate. For $x \in \mathbb{R}$, let $\lceil x \rceil$ denote the smallest integer greater than or equal to x, let $\lfloor x \rfloor$ denote the largest integer less than or equal to x, let $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$ denotes the nearest integer to x. For any $x, y \in \mathbb{R}$ with $y > 0$ we define $x \bmod y$ to be $x - \lfloor x/y \rfloor y$.

The (i, j)-th entry of a 2 dimensional matrix $\mathbf{M}$ is denoted by $m_{i,j}$. For any $i, j \in \mathbb{Z}$ such that $i < j$, denote by $[i, j]$ the set of integers $\{i, i+1, \cdots j-1, j\}$. For any positive integer a, denote by $[a]$ the set of integers $\{1, \cdots, a\}$, let $\mathbb{Z}_a$ denote $\mathbb{Z}/a\mathbb{Z}$, the elements of which are represented, by default, as $[0, a-1]$. We define $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, i.e., the group of reals $[0, 1)$ with modulo 1 addition.

If S is a finite set then $|S|$ is its cardinality, and $x \leftarrow S$ is the operation of choosing an element randomly from S. For any random variable X over $\mathbb{R}$, denote $\mathrm{supp}(X) = \{x \in \mathbb{R} | \Pr[X = x] > 0\}$. We use standard notations and conventions below for writing probabilistic algorithms and experiments. For a probability distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes the operation of choosing an element according to $\mathcal{D}$.

We use standard asymptotic (O, o, Ω, ω) notation to denote the growth of positive functions. We say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \log^c n)$ for some constant c. Let λ denote the security parameter (for constructions and analyses of LWE-based scheme, we also use the dimension, denoted l, of the underlying matrix as the security parameter). We say that a function $f(\lambda)$ is *negligible*, if for every $c > 0$, there exists a λ_c, such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$. For two distribution ensembles $\{X(\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$ and $\{Y(\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$, we say that they are computationally indistinguishable, denoted $\{X(\lambda, z)\} \overset{c}{\approx} \{Y(\lambda, z)\}$, if for any probabilistic polynomial-time (PPT) algorithm D, and for sufficiently large λ and any $z \in \{0, 1\}^*$, it holds $|\Pr[D(\lambda, z, X)] = 1 - \Pr[D(\lambda, z, Y)] = 1|$ is *negligible* in λ.

2.1. Definitions

In this section, we recall the definitions of cryptographic primitives and cryptosystems, lossy trapdoor functions (LTDF) and all-but-one trapdoor functions (ABO-TDF). Then we sketch out and summarize in Section 8 the constructions of these primitives and cryptosystems from LTDF and ABO-TDF according to the results in [8].

Definitions of Cryptographic Primitives and Cryptosystems. We recall the definitions of injective trapdoor functions, collision resistant and universal one-way hash functions, strongly unforgeable one-time signature, and public-key encryption (including security under chosen-plaintext attack (CPA) and under chosen-ciphertext attack(CCA)). Let $n = n(\lambda) = \mathrm{poly}(\lambda)$ denote the input length of the trapdoor functions.

Definition 1. A collection of injective trapdoor functions is described by a tuple of PPT algorithms (S, F, F^{-1}), having the following properties:

1. *Easy to sample, compute, and invert with trapdoor:* S outputs (s, t) where s is a function index and t is

its trapdoor, $F(s,\cdot)$ computes an injective function $f_s(\cdot)$ over the domain $\{0,1\}^n$, and $F^{-1}(t,\cdot)$ computes $f_s^{-1}(\cdot)$.

2. *Hard to invert without trapdoor:* for any PPT inverter $\mathcal{A}$, the probability that $\mathcal{A}(s, f_s(x))$ outputs x is negligible, where the probability is taken over the choice of $(s,t) \leftarrow S$, $x \leftarrow \{0,1\}^n$, and $\mathcal{A}$'s randomness.

Definition 2. A collection of collision-resistant hash functions (CRHFs) from length $\ell(\lambda)$ to length $\ell'(\lambda) < \ell(\lambda)$ is given by a pair of PPT algorithms $(S_{\text{crh}}, F_{\text{crh}})$, where

1. S_{crh} outputs a function index i;
2. $F_{\text{crh}}(i,\cdot)$ computes a function $H_i : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{\ell'(\lambda)}$;
3. for every PPT adversary $\mathcal{A}$, the probability over the choice of i and the randomness of $\mathcal{A}$ that $\mathcal{A}(i)$ outputs distinct $x, x' \in \{0,1\}^{\ell(\lambda)}$ such that $H_i(x) = H_i(x')$ is negligible in λ.

Definition 3. A collection of universal one-way hash functions (UOWHFs) from length $\ell(\lambda)$ to length $\ell'(\lambda) < \ell(\lambda)$ is given by a pair of PPT algorithms $(S_{\text{uowhf}}, F_{\text{uowhf}})$, where

1. S_{uowhf} outputs a function index i;
2. $F_{\text{uowhf}}(i,\cdot)$ computes a function $H_i : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{\ell'(\lambda)}$;
3. for every PPT adversary $\mathcal{A}$, the probability over the choice of i and the randomness of $\mathcal{A}$ that $\mathcal{A}(i)$ outputs some $x' \in \{0,1\}^{\ell(\lambda)}$ such that $x' \neq x$ and $H_i(x) = H_i(x')$ is negligible in λ.

Definition 4. A signature scheme is described by a tuple of PPT algorithms (Gen, Sign, Ver), which is modeled as follows:

- Gen outputs a signing key sk_σ and a verification key vk.
- $\mathsf{Sign}(sk_\sigma, m)$ takes as input a signing key sk_σ and a message $m \in \mathcal{M}$ (where $\mathcal{M}$ is some fixed message space) and outputs a signature σ.
- $\mathsf{Ver}(vk, m, \sigma)$ takes as input a verification key vk, a message $m \in \mathcal{M}$ and a signature σ, and outputs either 0 or 1.

For any $(sk_\sigma, vk) \leftarrow \mathsf{Gen}$ and any $m \in \mathcal{M}$, we require $\mathsf{Ver}(vk, m, \mathsf{Sign}(sk_\sigma, m)) = 1$ for *completeness*. We can also relax this notion to require that Ver outputs 1 with overwhelming probability over all the randomness of the experiment.

We define the security notion of *strong* existential unforgeability under a *one-time* chosen message attack by describing an experiment between a challenger and a PPT adversary algorithm $\mathcal{A}$ as follows: First, the challenger generates a key pair $(sk_\sigma, vk) \leftarrow \mathsf{Gen}$, and gives vk to $\mathcal{A}$. Then $\mathcal{A}$ may query an oracle that computes $\mathsf{Sign}(sk_\sigma,\cdot)$ on a single message $m \in \mathcal{M}$ of its choice, receiving a signature σ. Finally, $\mathcal{A}$ outputs a pair (m', σ'). If $\mathsf{Ver}(vk, m', \sigma') = 1$, then $\mathcal{A}$ wins. If a signature query was made by $\mathcal{A}$, we restrict $(m', \sigma') \neq (m, \sigma)$. The advantage of $\mathcal{A}$ is the probability (over all the randomness of the experiment) that $\mathcal{A}$ wins. We say that a signature scheme is strongly unforgeable under a one-time chosen message attack if no PPT adversary $\mathcal{A}$ can win the above game with non-negligible probability.

Definition 5. A cryptosystem is described by a tuple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ that are modeled as follows:

- $\mathcal{G}$ outputs a public key pk and a secret key sk.
- $\mathcal{E}(pk, m)$ on input a public key pk and a message $m \in \mathcal{M}$ (where $\mathcal{M}$ is some message space), and outputs a ciphertext c.
- $\mathcal{D}(sk, c)$ on input a secret key sk and a ciphertext c, and outputs a message $m \in \mathcal{M} \cup \{\perp\}$, where $\perp$ is a distinguished symbol indicating decryption failure.

For any $(pk, sk) \leftarrow \mathcal{G}$ and any $m \in \mathcal{M}$, we require $\mathcal{D}(sk, \mathcal{E}(pk, m)) = m$ for *completeness*. We can also relax this notion to require that decryption is correct with overwhelming probability over all the randomness of the experiment.

We say that a public key cryptosystem is *CPA security*, if it is indistinguishability under a chosen plaintext attack. That is, the views of any PPT adversary $\mathcal{A}$ in the following two experiments indexed by a bit $b \in \{0,1\}$ are computationally indistinguishable: a key pair $(pk, sk) \leftarrow \mathcal{G}$ is generated and pk is given to $\mathcal{A}$. Then $\mathcal{A}$ outputs two messages $m_0, m_1 \in \mathcal{M}$, and is given a ciphertext $c^* \leftarrow \mathcal{E}(pk, m_b)$, i.e., an encryption of message m_b.

We say that a public key cryptosystem is *CCA security*, if it is indistinguishability under an adaptive chosen ciphertext attack. This notion is similarly defined by two experiments as described above, the difference is that the adversary $\mathcal{A}$ is additionally given access to an oracle $\mathcal{O}$ that computes $\mathcal{D}(sk,\cdot)$ during part or all of the game. If the adversary $\mathcal{A}$ can only access to the oracle $\mathcal{O}$ *before* the ciphertext c^* is given to $\mathcal{A}$, we call it *CCA1 security*. If the oracle $\mathcal{O}$ computes $\mathcal{D}(sk,\cdot)$ throughout the entire experiment, with the exception that it returns $\perp$ if queried on the particular challenge ciphertext c^*, we call it *CCA2 security*.

Definitions of LTDF and ABO-TDF. Here we describe the notions of lossy trapdoor functions (LTDF), and all-but-one trapdoor functions (ABO-TDF).

Let $n(\lambda) = \text{poly}(\lambda)$ denote the input length of the function, and let $k(\lambda) \leq n(\lambda)$ denote the *lossiness* of the collection. For presentation simplicity, we usually omit the dependence on λ for convenience.

Definition 6. A collection of (n, k)-*lossy trapdoor functions* is described by a tuple of (possibly probabilistic) polynomial-time algorithms $(S_{\text{ltdf}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$, having the properties below.

1. *Easy to sample an injective function with trapdoor:* $S_{\text{inj}}(1^\lambda)$ outputs (s, t) where s is a function index and t is its trapdoor. $F_{\text{ltdf}}(s, \cdot)$ computes a (deterministic) *injective* function $f_s(\cdot)$ over the domain $\{0,1\}^n$, and $F^{-1}_{\text{ltdf}}(t, \cdot)$ computes $f_s^{-1}(\cdot)$. If a value y is not in the image $f_s(\{0,1\}^n)$, i.e., if $f_s^{-1}(y)$ does not exist, then the behavior of $F^{-1}_{\text{ltdf}}(t, y)$ is unspecified. Note that some applications may need to check the output of F^{-1}_{ltdf} for correctness.

2. *Easy to sample a lossy function:* $S_{\text{loss}}(1^\lambda)$ outputs $(s, \perp)$ where s is a function index, and $F_{\text{ltdf}}(s, \cdot)$ computes a (deterministic) function $f_s(\cdot)$ over the domain $\{0,1\}^n$ whose image has size at most 2^{n-k}.

3. *Hard to distinguish injective from lossy:* the first outputs of $S_{\text{inj}}(1^\lambda)$ and $S_{\text{loss}}(1^\lambda)$ are computational indistinguishable. More formally, letting X_λ denote the distribution of s from $S_{\text{inj}}(1^\lambda)$, and letting Y_λ denote the distribution of s from $S_{\text{loss}}(1^\lambda)$, then $\{X_\lambda\} \overset{c}{\approx} \{Y_\lambda\}$.

As shown in [8], for constructing lattice-based LTDF, a slightly relaxed definition of lossy TDF is considered, which is called *almost-always* lossy TDF. That is, the output of S_{inj} describes an injective function f_s that F^{-1}_{ltdf} inverts correctly on all values in the image of f_s *with overwhelming probability*. Namely, the probability (over the choice of s) that f_s is not injective or that $F^{-1}_{\text{ltdf}}(t, \cdot)$ computes $f_s^{-1}(\cdot)$ on some input incorrectly is *negligible*. Moreover, the image size of the lossy function f_s generated by S_{loss} is required to be, with overwhelming probability, at most 2^{n-k}. In general, the function sampler cannot check these conditions (i.e., whether $f_s(\cdot)$ is injective , or whether $F^{-1}_{\text{ltdf}}(t, \cdot)$ correctly computes $f_s^{-1}(\cdot)$ for all input), because they are associated with global probabilities of the generated function. Since the generation of trapdoor/lossy functions does not under the control of the adversary, we may make use of almost-always lossy TDF without affecting security of all the applications (e.g., CCA-secure encryption), and the potential advantage of the adversary due to sampling an improper function is bounded by a negligible quantity.

The combination of the lossiness and indistinguishability properties implies that the injective function is one-wayness, as shown in the following lemma given in [8].

Lemma 1. Let $(S_{\text{ltdf}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$ give a collection of (n, k)-LTDF with $k \geq \omega(\log \lambda)$. Then $(S_{\text{inj}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$ gives a collection of injective trapdoor function. (The analogous result applies for almost-always collections.)

In order to construct CCA-secure cryptosystem from LTDF, it is more convenient to consider a new notion called *all-but-one* trapdoor function (ABO-TDF) [8]. In an ABO-TDF collection, each function has multiple branches. One branch is lossy, while (super-polynomially) many others are injective trapdoor functions owning the same trapdoor.

Definition 7. Let $\mathcal{B} = \{B_\lambda\}_{\lambda \in \mathbb{N}}$ denote a collection of sets whose elements represent the branches. A collection of (n, k)-*all-but-one* trapdoor functions with branch collection $\mathcal{B}$ is described by a tuple of (possible probabilistic) polynomial-time algorithms $(S_{\text{abo}}, G_{\text{abo}}, G^{-1}_{\text{abo}})$, having the following properties:

1. *Sampling a trapdoor function with given lossy branch:* for any $b^* \in B_\lambda$, $S_{abo}(1^\lambda, b^*)$ outputs (s, t), where s is a function index and t is its trapdoor.

   For any $b \in B_\lambda$ distinct from b^*, $G_{\text{abo}}(s, b, \cdot)$ computes a (deterministic) injective function $g_{s,b}(\cdot)$ over the domain $\{0,1\}^n$, $G^{-1}_{\text{abo}}(t, b, \cdot)$ computes $g^{-1}_{s,b}(\cdot)$. As above, the behavior of $G^{-1}_{\text{abo}}(t, b, y)$ is unspecified if $g^{-1}_{s,b}(y)$ does not exist.

   Additionally, $G_{\text{abo}}(s, b^*, \cdot)$ computes a function $g_{s,b^*}(\cdot)$ on the domain $\{0,1\}^n$ whose image has size at most 2^{n-k}.

2. *Hidden lossy branch:* for any $b_0^*, b_1^* \in B_\lambda$, the first output s_0 of $S_{abo}(1^\lambda, b_0^*)$ and the first output s_1 of $S_{abo}(1^\lambda, b_1^*)$ are computationally indistinguishable.

Similar to LTDF, for lattice-based constructions we consider almost-always ABO-TDF [8], i.e., the injective, invertible, and lossy properties are required to hold only with overwhelming probability over the choice of the function index s.

Remark 1. The basic relation between the two notions is revealed in [8]: lossy and ABO trapdoor functions are equivalent if we choose parameters appropriately. The reader is referred to [8] for more details.

2.2. Probability Distributions

We present the notions of *normal distribution* over $\mathbb{R}$, the *discrete distribution* over $\mathbb{Z}_q$, and a standard *tail inequality*. Given a positive real number $\sigma > 0$, the *normal distribution* with mean 0 and variance σ^2 (or standard deviation σ) is the distribution having density function $\rho_\sigma(x) = \exp(-x^2/2\sigma^2)/\sqrt{2\pi\sigma^2}$ for $x \in \mathbb{R}$. In fact, the sum of two independent normal variables with

mean 0 and variances σ_1^2 and σ_2^2, respectively, is a normal variable with mean 0 and variance $\sigma_1^2 + \sigma_2^2$.

For a positive real number $\alpha > 0$, we define Ψ_α to be the distribution on $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. For any probability distribution $\phi : \mathbb{T} \to \mathbb{R}_{>0}$ and a positive integer $q > 0$, we define its *discretization* $\bar{\phi} : \mathbb{Z}_q \to \mathbb{R}_{>0}$ to be the *discrete distribution* over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_\phi \rceil \mod q$, where X_ϕ has distribution ϕ.

For a positive real number $\sigma > 0$ and $t \geq 1$, let X be a normal variable with variance σ^2, a standard *tail inequality* tells that $\Pr[\,|X| < t\sigma\,] \geq 1 - \exp(-t^2)$.

2.3. The Learning with Errors Problem

The *learning with errors* (LWE) problem is a classic hard lattice problem proposed in [21] . The LWE problem can be viewed as an average-case "unique encoding" on a certain family of random lattices under a natural error distribution, and is believed to be hard on the average even against quantum computer. The following is almost verbatim from [21] and [8].

On input security parameter λ, for positive integers l and q, a vector $\mathbf{s} \in \mathbb{Z}_q^l$ and some probability distribution χ on $\mathbb{Z}_q$, let $A_{q,\mathbf{s},\chi}$ be the distribution over $\mathbb{Z}_q^l \times \mathbb{Z}_q$, obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^l$ uniformly at random as well as $e \leftarrow \chi$ independently, outputting the pair $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where all the above are operated in $\mathbb{Z}_q$. The error distribution χ is taken to be the discrete distribution as specified in Section 2.2.

The goal of the (decisional) *learning with errors* problem $\mathrm{LWE}_{q,\chi}$ in dimension l is to distinguish the distribution $A_{q,\mathbf{s},\chi}$ for some secret random $\mathbf{s} \leftarrow \mathbb{Z}_q^l$ from the uniform distribution over $\mathbb{Z}_q^l \times \mathbb{Z}_q$ with non-negligible probability, even if the adversary sees polynomially many samples and even if the secret vector $\mathbf{s}$ is drawn randomly from χ^l [24].

The dimension l is the main parameter for the hardness of LWE. In the rest of this paper, for the constructions and analysis of LWE-based schemes we simply let l instead of λ be the security parameter, and let other parameters like q, α, n, etc., be function of l. When $\alpha q \geq 2\sqrt{l}$, this decision problem is at least as hard as approximating several problems on l-dimensional lattices in the worst-case to within $\widetilde{O}(l/\alpha)$ factors with a *quantum* algorithm [21] , or via a *classical* algorithm for a subset of these problems [25]. We state a fact from [21] below:

Proposition 1. Let $\alpha = \alpha(l) \in (0, 1)$ and let $q = q(l)$ be a prime such that $\alpha \cdot q > 2\sqrt{l}$. If there exists an efficient (possibly quantum) algorithm such that solves $\mathrm{LWE}_{q,\Psi_\alpha}$, then there exists an efficient quantum algorithm for solving the following worst-case lattice problems:

- SIVP: In any lattice Λ of dimension l, find a set of l linearly independent lattice vectors of length within at most $\tilde{O}(l/\alpha)$ of optimal.
- GapSVP: In any lattice Λ of dimension l, approximate the length of a shortest nonzero lattice vector to within a $\tilde{O}(l/\alpha)$ factor.

In fact, to obtain a poly(l) approximate factor, known algorithms require time and space that are exponential in l [26], while known polynomial-time algorithms obtain approximation factors that are slightly subexponential in l [27, 28]. Thus SIVP and GapSVP problems appear to be quite hard in the worst case even for quantum algorithms.

According to [29], for lattice problem in any ℓ_p norm, where $2 < p \leq \infty$, the proposition still holds for substantially the same $\tilde{O}(l/\alpha)$ approximation factors [30]. Moreover, there is a *classical* reduction from a variant of the GapSVP problem to LWE for $\alpha q \geq \sqrt{l \log l}$ [25].

In the following, we construct our compact lossy trapdoor functions in terms of LWE problem, without considering the connection to lattices or the restrictions of the parameters. Later in Section 6, we will instantiate the parameters properly to invoke Proposition 1 to guarantee security, assuming the quantum worst-case hardness of lattice problems.

3. Compact (Homomorphic) Symmetric Encryption Scheme Based on LWE

We now construct compact symmetric encryption scheme based on the hardness of the LWE problem. This basic scheme has certain limited homomorphic properties over a small message space, which is enough for the purpose of constructing LTDF.

3.1. Encrypting Elements.

The message space is $\mathbb{Z}_p$ for some $p \geq 2$. For every message $m \in \mathbb{Z}_p$, define $c_m = \frac{m}{p} \in \mathbb{T}$. Let $q > p$ and $g \geq 2$ be integers, and let χ denote an unspecified error distribution that we will instantiate later. The scheme is as follows:

- $\mathbf{Gen}(1^l)$: The secret key is a uniform $\mathbf{s} \leftarrow \mathbb{Z}_q^l$.
- $\mathbf{Enc}(m \in \mathbb{Z}_p)$: It chooses uniform $\mathbf{a} \leftarrow \mathbb{Z}_q^l$ and an error term $e \leftarrow \chi$. Denote $\hat{c}_m = \langle \mathbf{a}, \mathbf{s} \rangle + e + \lfloor q c_m \rceil \bmod q \in \mathbb{Z}_q$. Define the *rounding errors*: $u = \lfloor q c_m \rceil - q c_m \in [-1/2, 1/2]$ and $u' = \lfloor g \hat{c}_m / q \rceil - g \hat{c}_m / q \in [-1/2, 1/2]$. The ciphertext is

$$E_{\mathbf{s}}(m, u, u'; \mathbf{a}, e) := (\mathbf{a}, g(\langle \mathbf{a}, \mathbf{s} \rangle + e + q c_m + u)/q + u') \quad (1)$$

  Here $E_{\mathbf{s}}(m, u, u'; \mathbf{a}, e) \in \mathbb{Z}_q^l \times \mathbb{Z}_g$. The reason that we treat u and u' as explicit input to the

encryption algorithm, even though they are usually determined by m, is that we can treat $E_{\mathbf{s}}(m, u, u'; \mathbf{a}, e)$ as a well-defined expression even for either $u \notin [-1/2, 1/2]$ or $u' \notin [-1/2, 1/2]$. We can also omit them and denote the ciphertext as

$$E_{\mathbf{s}}(m; \mathbf{a}, e) := (\mathbf{a}, \lfloor g(\langle \mathbf{a}, \mathbf{s} \rangle + e + \lfloor qc_m \rceil)/q \rceil). \quad (2)$$

- **Dec**$(\mathbf{s}, c)$: For $c = (\mathbf{a}, c')$, compute

$$m' = \lfloor p(c'/g - \langle \mathbf{a}, \mathbf{s} \rangle/q) \rceil \bmod p. \quad (3)$$

Proposition 2. The above encryption scheme is correct.

Proof. For any ciphertext $c = E_{\mathbf{s}}(m, u, u'; \mathbf{a}, e)$, we have

$$\begin{aligned} m' &= \lfloor p(c'/g - \langle \mathbf{a}, \mathbf{s} \rangle/q) \rceil \bmod p & (4)\\ &= \left\lfloor m + \frac{p}{q}(\langle \mathbf{a}, \mathbf{s} \rangle + e - \langle \mathbf{a}, \mathbf{s} \rangle) + \frac{p}{q}u + \frac{p}{g}u' \right\rfloor \bmod p & (5)\\ &= \left\lfloor m + \frac{p}{q}e + \frac{p}{q}u + \frac{p}{g}u' \right\rfloor \bmod p. & (6) \end{aligned}$$

As long as the absolute $|pe/q + pu/q + pu'/g| \le p|e|/q + p/2q + p/2g < 1/2$, i.e., $(2|e| + 1)p < q(1 - \frac{p}{g})$, the decryption **Dec**$(c, \mathbf{s})$ is correct. □

Proposition 3. The above scheme is homomorphic.

Proof. By a simple calculation, we have

$$\begin{aligned} &E_{\mathbf{s}}(m_1, u_1, u_1'; \mathbf{a}_1, e_1) + E_{\mathbf{s}}(m_2, u_2, u_2'; \mathbf{a}_2, e_2) & (7)\\ =&E_{\mathbf{s}}(m_1 + m_2, u_1 + u_2, u_1' + u_2'; \mathbf{a}_1 + \mathbf{a}_2, e_1 + e_2). & (8) \end{aligned}$$

Furthermore, even without knowing the secret key under which a ciphertext was created, one can add any scalar value $v \in \mathbb{Z}_p$ to its plaintext. Let $c = (\mathbf{a}, c') = E_{\mathbf{s}}(m, u, u'; \mathbf{a}, e)$, define $u'' = \lfloor qc_v \rceil - qc_v \in [-1/2, 1/2]$ and $u''' = \lfloor g\lfloor qc_v \rceil/q \rceil - g\lfloor qc_v \rceil/q \in [-1/2, 1/2]$, then

$$\begin{aligned} c \boxplus v &:= (\mathbf{a}, c' + \lfloor g\lfloor qc_v \rceil/q \rceil) & (9)\\ &= E_{\mathbf{s}}(m + v, u + u'', u' + u'''; \mathbf{a}, e). & (10) \end{aligned}$$

□

3.2. Encrypting Matrices.

The message space is $\mathbb{Z}_p^{h \times w}$ for arbitrary positive integers h and w. For every message $\mathbf{M} = (m_{i,j}) \in \mathbb{Z}_p^{h \times w}$, we describe an extension of the symmetric encryption scheme from encrypting elements to encrypting matrices.

- **Gen**(1^l): For every column $j \in [w]$, choose independently $\mathbf{s}_j \in \mathbb{Z}_q^l$. The secret key is the tuple $\mathbf{S} = (\mathbf{s}_1, \cdots, \mathbf{s}_w)$.
- **Enc**$(\mathbf{M} \in \mathbb{Z}_p^{h \times w})$: For every row $i \in [h]$, choose independently $\mathbf{a}_i \leftarrow \mathbb{Z}_q^l$, and forming a matrix $\mathbf{A} \in \mathbb{Z}_q^{h \times l}$ whose i-th row is $\mathbf{a}_i$. For every $i \in [h]$ and every $j \in [w]$, choose independently error terms $e_{i,j} \leftarrow \chi$, forming an error matrix $\mathbf{E} = (e_{i,j}) \in \mathbb{Z}_q^{h \times w}$. Denote $\hat{c}_{m_{i,j}} = \langle \mathbf{a}_i, \mathbf{s}_j \rangle + e_{i,j} + \lfloor qc_{m_{i,j}} \rceil \bmod q \in \mathbb{Z}_q$. Define $\mathbf{U} = (u_{i,j})$ and $\mathbf{U}' = (u'_{i,j})$ to be matrices of rounding errors, where $u_{i,j} = \lfloor qc_{m_{i,j}} \rceil - qc_{m_{i,j}} \in [-\frac{1}{2}, \frac{1}{2}]$ and $u'_{i,j} = \lfloor g\hat{c}_{m_{i,j}}/q \rceil - g\hat{c}_{m_{i,j}}/q \in [-\frac{1}{2}, \frac{1}{2}]$. The encryption of $\mathbf{M}$ is

$$\mathbf{C} = E_{\mathbf{S}}(\mathbf{M}, \mathbf{U}, \mathbf{U}'; \mathbf{A}, \mathbf{E}), \quad (11)$$

where $c_{i,j} = E_{\mathbf{s}_j}(m_{i,j}, u_{i,j}, u'_{i,j}; \mathbf{a}_i, e_{i,j})$. Note that the i-th row shares the same randomness $\mathbf{a}_i$, while the j-th column shares the same secret key $\mathbf{s}_j$. The ciphertext can be expressed as $(\mathbf{A}, \mathbf{C}')$, where $c'_{i,j} = g(\langle \mathbf{a}_i, \mathbf{s}_j \rangle + e_{i,j} + qc_{m_{i,j}} + u_{i,j})/q + u'_{i,j}$.

- **Dec**$(\mathbf{S}, \mathbf{C})$: For $\mathbf{C} = (c_{i,j})$, the decrypted matrix is $\mathbf{M} = (m_{i,j}) \in \mathbb{Z}_p^{h \times w}$, where $m_{i,j} = \mathbf{Dec}(\mathbf{s}_j, c_{i,j})$.

Correctness: The correctness is direct from that of the basic scheme for encrypting elements.

Homomorphism: All linear operations, including addition of ciphertexts, multiplication and addition by scalars, can be extended to encrypted matrices based on the homomorphism of the underlying symmetric encryption scheme of elements.

In particular, for any $\mathbf{x} \in \mathbb{Z}_p^h$, for an encryption $\mathbf{C} = E_{\mathbf{S}}(\mathbf{M}, \mathbf{U}, \mathbf{U}'; \mathbf{A}, \mathbf{E})$ of some $\mathbf{M} \in \mathbb{Z}_p^{h \times w}$, we have

$$\mathbf{xC} = E_{\mathbf{S}}(\mathbf{xM}, \mathbf{xU}, \mathbf{xU}'; \mathbf{xA}, \mathbf{xE}). \quad (12)$$

Furthermore, for any matrix of scalars $\mathbf{V} \in \mathbb{Z}_p^{h \times w}$ inducing two matrices of rounding errors $\mathbf{U}''$ and $\mathbf{U}'''$, we have

$$\mathbf{C} \boxplus \mathbf{V} = E_{\mathbf{S}}(\mathbf{M} + \mathbf{V}, \mathbf{U} + \mathbf{U}'', \mathbf{U}' + \mathbf{U}'''; \mathbf{A}, \mathbf{E}). \quad (13)$$

Lemma 2. For any height and width $h, w = \mathrm{poly}(l)$, the matrix encryption scheme described above produces indistinguishable ciphertexts under the assumption that $\mathrm{LWE}_{q,\chi}$ is hard.

Proof. The proof is almost the same as Lemma 6.2 in [8], and we omit details here. □

Lemma 3. For some positive integer r and α, let $\mathbf{E} = (e_{i,j}) \in \mathbb{Z}_q^{n \times w}$ be an error matrix generated by choosing independent error terms $e_{i,j} \leftarrow \bar{\Psi}_\alpha$. Then except with probability at most $w \cdot 2^{-r}$ over the choice of $\mathbf{E}$, every entry of $\mathbf{xE}$ has absolute value less than $(n + r)\alpha q + n/2$ for all $\mathbf{x} \in \{0, 1\}^n$.

Proof. The proof is easily extended from that in [29], so we omit it. □

Remark 2. With our compact encryption scheme, when encrypting an element m, the resulting ciphertext is $E_{\mathbf{s}}(m;\mathbf{a},e) = (\mathbf{a}, \lfloor g(\langle \mathbf{a},\mathbf{s}\rangle + e + \lfloor qc_m \rceil)/q \rceil) \in \mathbb{Z}_q^l \times \mathbb{Z}_g$, while the ciphertext is $E_{\mathbf{s}}(m;\mathbf{a},e) = (\mathbf{a}, \langle \mathbf{a},\mathbf{s}\rangle + e + \lfloor qc_m \rceil) \in \mathbb{Z}_q^l \times \mathbb{Z}_q$ in [8], where $q \approx gO(n^c)$, $c > 0$ is a constant. The length of our compact LTDF ciphertext is $\log q - \log g$ bits shorter than that of the scheme given in [8].

Similarly, when encrypting an matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times m}$, the ciphertext in this paper is $E_{\mathbf{S}}(\mathbf{M};\mathbf{A},\mathbf{E}) \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_g^{n\times m}$, which reduces $nm\log(q/g)$-bit length than that of $E_{\mathbf{S}}(\mathbf{M};\mathbf{A},\mathbf{E}) \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^{n\times m}$ in [8], where $q \approx gO(n^c)$, $c > 0$ is a constant and $m = n/\lfloor \log p \rfloor$.

4. Compact LTDF Based on LWE

Let $a = \lfloor \lg p \rfloor$, assume without loss of generality that n is divisible by a, and let $m = n/a$. Define a matrix $\mathbf{G} \in \mathbb{Z}_p^{n\times m}$ as follows: in column $j \in [m]$, the $((j-1)a+k)$th entry is $2^{k-1} \in [1,p]$ for $k \in [a]$. All other entries are zero. Formally, $\mathbf{G}$ is the tensor product $\mathbf{I}_m \otimes \mathbf{g}$, where $\mathbf{I}_m$ is the identity matrix and $\mathbf{g} = (1,2,\cdots,2^{a-1})^T \in \mathbb{Z}_p^{a\times 1}$ (we can also use other integer base $b \geq 2$).

For any input vector $\mathbf{x} \in \{0,1\}^n$, we may correspond $\mathbf{x}$ to an unique vector $\mathbf{v} = (v_1,\cdots,v_m) \in \mathbb{Z}_p^m$ using the matrix $\mathbf{G}$; That is, $\mathbf{xG} = \mathbf{v}$, and vice versa.

Evaluating the function on $\mathbf{x} \in \{0,1\}^n$ involves homomorphically computing an encrypted linear product $\mathbf{xM}$, where $\mathbf{M}$ is some matrix being encrypted in the sampling algorithm. In the injective case, let $\mathbf{M} = \mathbf{G}$, then $\mathbf{xG} = \mathbf{v}$, which allows us to recover the entire input by decrypting $\mathbf{v}$ and producing the corresponding $\mathbf{x}$. In the lossy case, we have $\mathbf{M} = \mathbf{0}$, then $\mathbf{xM} = \mathbf{0} \in \mathbb{Z}_p^m$, which means the output contains only $m = n/a$ ciphertexts, i.e., less information is leaked via the error terms.

In order to obtain a lossy TDF, we need to ensure that each decrypted plaintext contains more information than what might be carried by the error terms of the corresponding ciphertext. In the following, we describe our lossy TDF generation, evaluation, and inversion algorithms formally.

- *Sampling an injective/lossy function.* The generator of injective function $S_{\text{inj}}(1^l)$ outputs a matrix encryption

$$\mathbf{C} = E_{\mathbf{S}}(\mathbf{G},\mathbf{U},\mathbf{U}';\mathbf{A},\mathbf{E}), \tag{14}$$

  where $\mathbf{S}$, $\mathbf{U}$, $\mathbf{U}'$, $\mathbf{A}$, $\mathbf{E}$ are chosen as described in Section 3.2. The function index s is the encryption $\mathbf{C}$, and the trapdoor information t consists of the tuple of secret keys $\mathbf{S} = (\mathbf{s}_1,\cdots,\mathbf{s}_m)$.
  The generator of lossy function $S_{\text{loss}}(1^l)$ generates a matrix encryption

$$\mathbf{C} = E_{\mathbf{S}}(\mathbf{0},\mathbf{U},\mathbf{U}';\mathbf{A},\mathbf{E}), \tag{15}$$

  which is the encryption of the all-zeros matrix $\mathbf{0} \in \mathbb{Z}_p^{n\times m}$. The function index s is $\mathbf{C}$, and there is no trapdoor output.

- *Evaluation algorithm.* On input $(\mathbf{C},\mathbf{x})$ where $\mathbf{C}$ is the function index (an encryption of either $\mathbf{M} = \mathbf{G}$ or $\mathbf{M} = \mathbf{0}$) and $\mathbf{x} \in \{0,1\}^n$ is an n-bit input interpreted as a vector, the evaluation function F_{ltdf} outputs the vector of ciphertexts $\mathbf{y} = \mathbf{xC}$.
  By the properties of homomorphism, the output $\mathbf{y}$ is

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{xM},\mathbf{xU},\mathbf{xU}';\mathbf{xA},\mathbf{xE}), \tag{16}$$

  where every ciphertext y_j is of the form $(\mathbf{xA}, y_j') \in \mathbb{Z}_q^l \times \mathbb{Z}_g$.

- *Inversion algorithm.* On input $(\mathbf{S},\mathbf{y})$ where $\mathbf{S}$ is the trapdoor, the inversion function F_{ltdf}^{-1} computes $\mathbf{v} = \mathbf{Dec}(\mathbf{S},\mathbf{y}) \in \mathbb{Z}_p^m$, and outputs the unique $\mathbf{x} \in \{0,1\}^n$ such that $\mathbf{xG} = \mathbf{v}$.

Similar to [8, 29], we now instantiate the parameters of the above scheme to prove that, conditioned on the assumption $\text{LWE}_{q,\chi}$ is hard, our construction describe a collection of almost-always (n,k)-lossy TDF. For simplicity, we assume modulus $p \geq 2$ is a power of 2 in the following theorem.

Theorem 1. Let $m = n/\lg p$, let $q \geq 4pn = 4mp\lg p$, let $g \in [4pn, q]$, where $p \geq 2$ is a power of 2, and let $\chi = \bar{\Psi}_\alpha$ where $\alpha \leq 1/(16pn) = 1/(16mp\lg p)$.

Then the algorithms described above give a collection of almost-always (n,k)-lossy TDF under the assumption that $\text{LWE}_{q,\chi}$ is hard, where the residual leakage $n-k$ is

$$n - k \leq n \cdot \left(\frac{l}{m} + (\frac{l}{m} + 1)\log_p(\frac{q}{p}) \right). \tag{17}$$

Note that in order for the residual leakage rate to be less than 1, we need both $m > l$ and $q < p^2$.

Proof. First we claim that the inversion algorithm F_{ltdf}^{-1} satisfies, with overwhelming probability over the choice of $\mathbf{C}$ by S_{inj}, the correctness requirement on all inputs $\mathbf{y} = F_{\text{ltdf}}(\mathbf{C},\mathbf{x})$. We note that

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{xM},\mathbf{xU},\mathbf{xU}';\mathbf{xA},\mathbf{xE}), \tag{18}$$

by the homomorphic properties.

Letting $r = n$ in Lemma 3, we have $|(\mathbf{xE})_j| < (n+r)\alpha q + n/2 \leq q/4p$ for every $\mathbf{x}$ and $j \in [m]$, except with probability at most $m \cdot 2^n = \text{negl}(l)$ over the choice of $\mathbf{E}$. Moreover, note that $|(\mathbf{xU})_j| \leq n/2 \leq q/8p$ and $|(\mathbf{xU}')_j| \leq n/2 \leq q/8p$ for all $j \in [m]$ by the size of $\mathbf{U}$'s and $\mathbf{U}'$'s entries. Therefore we have

$$|\frac{p}{q}(\mathbf{xE})_j + \frac{p}{q}(\mathbf{xU})_j + \frac{p}{g}(\mathbf{xU}')_j| \tag{19}$$

$$< \frac{p}{q}\cdot\frac{q}{4p} + \frac{p}{q}\cdot\frac{q}{8p} + \frac{p}{g}\cdot\frac{n}{2} \tag{20}$$

$$\leq \frac{1}{4} + \frac{1}{8} + \frac{1}{8} = \frac{1}{2}. \tag{21}$$

The correctness requirement is satisfied.

We now analyze the lossiness of a lossy function. For any input $\mathbf{x} \in \{0,1\}^n$, we have

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{0} = \mathbf{x0}, \mathbf{xU}, \mathbf{xU}'; \mathbf{xA}, \mathbf{xE}). \tag{22}$$

For every $j \in [m]$, y_j is a ciphertext $(\mathbf{xA}, y'_j) \in \mathbb{Z}_q^l \times \mathbb{Z}_g$, where $\mathbf{xA}$ is the same randomness for all j and $y'_j = g(\langle \mathbf{xA}, \mathbf{s}_j \rangle + (\mathbf{xE})_j + 0 + 0)/q + (\mathbf{xU}')_j)$. Fixing $\mathbf{A}, \mathbf{x}$ and $j \in [m]$, we have

$$|\frac{g}{q}(\mathbf{xE})_j + (\mathbf{xU}')_j| < \frac{g}{q} \cdot \frac{q}{4p} + \frac{n}{2} \leq \frac{q}{4p} + \frac{q}{8p} = \frac{q}{2p}. \tag{23}$$

Obviously, the total number of outputs of the lossy function is at most $q^l(q/p)^m$, the logarithm of which gives an upper bound on the residual leakage $n - k$:

$$\begin{aligned} n - k &\leq l \cdot \lg q + m \cdot \lg(\frac{q}{p}) && (24)\\ &= n \cdot \frac{l \lg q}{m \lg p} + \frac{n}{\lg p} \cdot \lg(\frac{q}{p}) && (25)\\ &= n \cdot \left(\frac{l}{m} + (\frac{l}{m} + 1) \log_p(\frac{q}{p}) \right). && (26)\\ & && (27) \end{aligned}$$

Finally, note that $\mathbf{C} = E_{\mathbf{S}}(\mathbf{G}, \mathbf{U}, \mathbf{U}'; \mathbf{A}, \mathbf{E})$ is indistinguishable from $\mathbf{C} = E_{\mathbf{S}}(\mathbf{0}, \mathbf{U}, \mathbf{U}'; \mathbf{A}, \mathbf{E})$ by Lemma 2 on the security of matrix encryption; That is, we can not distinguish lossy function from injective one. □

5. Compact ABO-TDF

In order to yield enough branches, the construction of ABO-TDF in [8] makes use of a family of pairwise independent hash functions $\mathcal{H} = \{h : \mathbb{Z}_p^l \to \mathbb{Z}_p^{m \times w}\}$, where $w = m + 2l$, to generate the matrix $\mathbf{M} = -h(b^*) \otimes \mathbf{g}$ for the desired branch b^*. The properties of the pairwise independent function h ensure that $\mathbf{H} = h(b) - h(b^*)$ have full row rank for any $b \neq b^*$, which suffices for recovering $\mathbf{v}$ from the product $\mathbf{vH}$. The branch set is $B = B_l = \mathbb{Z}_p^l$. Note that, with this approach, the encrypted matrix of function indices of ABO-TDF is larger than that of LTDF in [8]. We are wondering whether we can further reduce it? The answer is yes. To solve this problem, we make use of the full rank difference (FRD) encoding [23] (instead of the pairwise independent hash function originally used in [8]), which not only reduces the matrix size in our ABO-TDF to get equal to that of our compact LTDF, but also can allow smaller system parameters to support super-polynomially many injective branches in the construction of CCA secure public key encryption. We first briefly review the full rank difference encoding technique proposed in [23].

5.1. Full Rank Difference Encoding $\mathsf{G}_{\mathrm{FRD}}$ of $\mathbb{Z}_p^m$ to $\mathbb{Z}_p^{m \times m}$

In fact, Cramer and Damgård [23] introduced an encoding function maps a superpolynomially-sized domain $\mathbb{F}^m$ to matrices in $\mathbb{F}^{m \times m}$ with some strongly injective properties. This encoding notion has then been updated by [31] to the name "Full-Rank Difference Encoding". We uses FRD in a similar way to [29] .

Definition 8. Let p be a prime and m a positive integer. We say that a function $\mathsf{G}_{\mathrm{FRD}} : \mathbb{Z}_p^m \to \mathbb{Z}_p^{m \times m}$ is an encoding with full-rank difference (FRD) if:

1. for all distinct $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^m$, the matrix $\mathsf{G}_{\mathrm{FRD}}(\mathbf{x}) - \mathsf{G}_{\mathrm{FRD}}(\mathbf{y})$ is full rank.
2. $\mathsf{G}_{\mathrm{FRD}}$ is computable in polynomial time.

The goal in designing $\mathsf{G}_{\mathrm{FRD}}$ is to construct an additive subgroup $\mathcal{G}$ of $\mathbb{Z}_p^{m \times m}$ of size p^m with all non-zero matrices in $\mathcal{G}$ of full rank. Since for all distinct $\mathbf{A}, \mathbf{B} \in \mathcal{G}$, the difference $\mathbf{A}$-$\mathbf{B}$ is also in $\mathcal{G}$, it follows that $\mathbf{A}$-$\mathbf{B}$ is full rank.

For a polynomial $g \in \mathbb{Z}_p[X]$ of degree at most $m-1$, let $\mathsf{coeff}(g) \in \mathbb{Z}_p^{1 \times m}$ be the m-row of the coefficient of g. If g is of degree less than $m-1$ we pad the coefficients vector with zeroes on the right to make it a m-vector. Let f be some polynomial of degree m, irreducible in $\mathbb{Z}_p[X]$. Note that for a polynomial $g \in \mathbb{Z}_p[X]$, the polynomial $(g \bmod f)$ has degree less than m, thus $\mathsf{coeff}(g \bmod f) \in \mathbb{Z}_p^m$.

For any integer $m \geq 2$, any input $\mathbf{h} = (h_0, \cdots, h_{m-1}) \in \mathbb{Z}_p^m$, define $g_{\mathbf{h}}(X) = \sum_{i=0}^{m-1} h_i x^i \in \mathbb{Z}_p[X]$, then define $\mathsf{G}_{\mathrm{FRD}}(\mathbf{h})$ as

$$\mathsf{G}_{\mathrm{FRD}}(\mathbf{h}) := \begin{bmatrix} \mathsf{coeff}(g_{\mathbf{h}} \bmod f) \\ \mathsf{coeff}(X \cdot g_{\mathbf{h}} \bmod f) \\ \mathsf{coeff}(X^2 \cdot g_{\mathbf{h}} \bmod f) \\ \vdots \\ \mathsf{coeff}(X^{(m-1)} \cdot g_{\mathbf{h}} \bmod f) \end{bmatrix} \in \mathbb{Z}_p^{m \times m}. \tag{28}$$

The following theorem in [23] proves that the above function $\mathsf{G}_{\mathrm{FRD}}$ is an FRD.

Theorem 2. Let $\mathbb{F}$ be a field and f a polynomial in $\mathbb{F}[X]$. If f is irreducible in $\mathbb{F}[X]$ then the function $\mathsf{G}_{\mathrm{FRD}}$ defined above is an encoding with full-rank differences.

Moreover, the function $\mathsf{G}_{\mathrm{FRD}}$ has the following properties:

1. ($\mathsf{G}_{\mathrm{FRD}}$ is linear) $\mathsf{G}_{\mathrm{FRD}}(a\mathbf{h}_1 + b\mathbf{h}_2) = a \cdot \mathsf{G}_{\mathrm{FRD}}(\mathbf{h}_1) + b \cdot \mathsf{G}_{\mathrm{FRD}}(\mathbf{h}_2)$ for any $a, b \in \mathbb{Z}_p, \mathbf{h}_1, \mathbf{h}_2 \in \mathbb{Z}_p^m$.
2. (The image of $\mathsf{G}_{\mathrm{FRD}}$ is invertible or zero) For any vector $\mathbf{h} \neq \mathbf{0}$, $\mathsf{G}_{\mathrm{FRD}}(\mathbf{h})$ is invertible, and $\mathsf{G}_{\mathrm{FRD}}(\mathbf{0}) = \mathbf{0}$.

5.2. Construction and Analysis of Compact ABO-TDF

As above, let $a = \lfloor \lg p \rfloor$, assume without loss of generality that n is divisible by a, and let $m = n/a$. Define a matrix $\mathbf{G} := \mathbf{I}_m \otimes \mathbf{g}$, where $\mathbf{I}_m$ is the identity matrix and $\mathbf{g} = (1, 2, \cdots, 2^{a-1})^T \in \mathbb{Z}_p^{a\times 1}$ (we can also use other integer base $b \geq 2$). Using matrix $\mathbf{G}$ allows us to correspond each the input vector $\mathbf{x} \in \{0,1\}^n$ to a unique vector $\mathbf{v} = (v_1, \cdots, v_m) \in \mathbb{Z}_p^m$ by $\mathbf{xG} = \mathbf{v}$.

In our construction, instead of using the family of pairwise independent hash functions, we consider the full rank difference encoding function $\mathsf{G}_{\mathrm{FRD}}$ described in Section 5.1, to generate the matrix $\mathbf{M} = -\mathsf{G}_{\mathrm{FRD}}(b^*) \otimes \mathbf{g}$ for the desired branch b^*. The properties of the FRD function guarantees that $\mathbf{H} = \mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*)$ is invertible for all $b^* \neq b$, which is enough for the purpose of recovering $\mathbf{v}$ from $\mathbf{vH}$. It turns out to be a larger branch set $B = \mathbb{Z}_p^m$ if we fix parameter p, where $m = n/\lfloor \lg p \rfloor > l$, according to the following instantiated parameters.This means that we can choose smaller p in order to support super-polynomially many injective branches in the construction of CCA secure public key encryption.

Evaluating the ABO function on an input $\mathbf{x} \in \{0,1\}^n$ involves computing an encrypted linear product $\mathbf{vM}$, where $\mathbf{M}$ is some matrix deciding by the branch b^* of the function being evaluated. The explicit fact that $\mathbf{x}(\mathbf{M} \otimes \mathbf{g}) = \mathbf{vM}$ for any $\mathbf{M} \in \mathbb{Z}_p^{m\times m}$ plays an important role in our construction of ABO-TDF. Let $\mathbf{M} = -\mathsf{G}_{\mathrm{FRD}}(b^*) \otimes \mathbf{g}$, then $\mathbf{x}((\mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*)) \otimes \mathbf{g}) = \mathbf{v}(\mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*))$, which allows us to recover the entire input by decrypting $\mathbf{v}(\mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*))$ and recovering $\mathbf{v}$, then producing the corresponding $\mathbf{x}$.

Let the branch set $B = \mathbb{Z}_p^m$. Let $\mathsf{G}_{\mathrm{FRD}}$ denote a full rank difference encoding from $B = \mathbb{Z}_p^m$ to $\mathbb{Z}_p^{m\times m}$ as introduced in Section 5.1. In the following, we describe our ABO-TDF generation, evaluation, and inversion algorithms formally.

- *Sampling an ABO function.* The function generator $S_{\mathrm{abo}}(1^l, b^* \in B)$ outputs a matrix encryption

$$\mathbf{C} = E_{\mathbf{S}}(-\mathsf{G}_{\mathrm{FRD}}(b^*) \otimes \mathbf{g}, \mathbf{U}, \mathbf{U}'; \mathbf{A}, \mathbf{E}), \tag{29}$$

  where $\mathbf{S}$, $\mathbf{U}$, $\mathbf{U}'$, $\mathbf{A}$, $\mathbf{E}$ are chosen as described in Section 3.2. The function index s is the encryption $\mathbf{C}$, and the trapdoor information t consists of the tuple of secret keys $\mathbf{S} = (\mathbf{s}_1, \cdots, \mathbf{s}_m)$ and the lossy branch value b^*.

- *Evaluation algorithm.* On input $(\mathbf{C}, b, \mathbf{x})$ where $\mathbf{C}$ is the function index, $b \in B$ is the desired branch, and $\mathbf{x} \in \{0,1\}^n$ is an n-bit input interpreted as a vector, the evaluation function G_{abo} outputs the vector of ciphertexts

$$\mathbf{y} = \mathbf{x}(\mathbf{C} \boxplus (\mathsf{G}_{\mathrm{FRD}}(b) \otimes \mathbf{g})). \tag{30}$$

  Let $\mathbf{H} = \mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*)$. Then by the properties of homomorphism, the output $\mathbf{y}$ is

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{x}(\mathbf{H} \otimes \mathbf{g}), \mathbf{x}(\mathbf{U} + \mathbf{U}''), \mathbf{x}(\mathbf{U}' + \mathbf{U}'''); \mathbf{xA}, \mathbf{xE}) \tag{31}$$

$$= E_{\mathbf{S}}(\mathbf{vH}, \mathbf{x}(\mathbf{U} + \mathbf{U}''), \mathbf{x}(\mathbf{U}' + \mathbf{U}'''); \mathbf{xA}, \mathbf{xE}) \tag{32}$$

  where $\mathbf{U}''$ and $\mathbf{U}'''$ are the matrices of rounding errors induced by the scalar matrix.

- *Inversion algorithm.* The function G_{abo}^{-1} takes as input $((\mathbf{S}, b^*), b, \mathbf{y})$, where $(\mathbf{S}, b^*)$ is the trapdoor information, b is the evaluated branch, and $\mathbf{y}$ is the function output. It first computes $\mathbf{m} = \mathbf{Dec}(\mathbf{S}, \mathbf{y}) \in \mathbb{Z}_p^m$. It then computes $\mathbf{H} = \mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*)$, if $\mathbf{H}$ is invertible, it computes $\mathbf{v} = \mathbf{mH}^{-1}$. Finally, it outputs the unique $\mathbf{x} \in \{0,1\}^n$ such that $\mathbf{xG} = \mathbf{v}$.

Theorem 3. Let $m = n/\lfloor \lg p \rfloor$, let $q \geq 20pn/3$, and let $g \in [20pn/3, q]$, where $p \geq 2$ is a prime. Let $\chi = \bar{\Psi}_\alpha$ where $\alpha \leq 1/(16pn) = 1/(16mp\lfloor \lg p \rfloor)$.

Then the algorithms described above give a collection of almost-always (n,k)-ABO-TDF with branch set $\mathbb{Z}_p^m$, under the assumption that $\mathrm{LWE}_{q,\chi}$ is hard, where the residual leakage $n - k$ is

$$n - k \leq n \cdot \left(\frac{l}{m} \cdot \frac{\lg p}{\lg p - 1} + (\frac{l}{m} + 1) \log_{\frac{p}{2}} (\frac{q}{p}) \right). \tag{33}$$

Proof. The proof is similar to that of Theorem 1. First we claim that the inversion algorithm G_{abo}^{-1} satisfies, with overwhelming probability over the choice of $\mathbf{C}$ by $S_{\mathrm{abo}}(1^l, b^*)$, the correctness requirement for all branches $b \neq b^*$ and on all inputs $\mathbf{y} = G_{\mathrm{abo}}(\mathbf{C}, b, \mathbf{x})$. We note that

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{vH} = \mathbf{x}(\mathbf{H} \otimes \mathbf{g}), \mathbf{x}(\mathbf{U} + \mathbf{U}''), \mathbf{x}(\mathbf{U}' + \mathbf{U}'''); \mathbf{xA}, \mathbf{xE}), \tag{34}$$

by the homomorphic properties. For every $j \in [m]$, y_j is a ciphertext $(\mathbf{xA}, y_j') \in \mathbb{Z}_q^l \times \mathbb{Z}_g$, where $\mathbf{xA}$ is the same randomness for all j and $y_j' = g(\langle \mathbf{xA}, \mathbf{s}_j \rangle + (\mathbf{xE})_j + q(\mathbf{vH})_j/p + (\mathbf{xU})_j + (\mathbf{xU}'')_j)/q + (\mathbf{xU}')_j + (\mathbf{xU}''')_j$.

Letting $r = n$ in Lemma 3, we have $|(\mathbf{xE})_j| < (n + r)\alpha q + n/2 \leq q/5p$ for every $\mathbf{x}$ and $j \in [m]$, except with probability at most $m \cdot 2^n = \mathrm{negl}(l)$ over the choice of $\mathbf{E}$. Moreover, note that $|(\mathbf{xU})_j| \leq n/2 \leq 3q/40p$ and so do $|(\mathbf{xU}')_j|, |(\mathbf{xU}'')_j|, |(\mathbf{xU}''')_j|$, for all $j \in [m]$ by the size of their entries. Therefore we have

$$|\frac{p}{q}(\mathbf{xE})_j + \frac{p}{q}((\mathbf{xU})_j + (\mathbf{xU}'')_j) + \frac{p}{g}((\mathbf{xU}')_j + (\mathbf{xU}''')_j)| \tag{35}$$

$$< \frac{p}{q} \cdot \frac{q}{5p} + \frac{p}{q} \cdot (\frac{3q}{40p} + \frac{3q}{40p}) + \frac{p}{g} \cdot (\frac{n}{2} + \frac{n}{2}) \tag{36}$$

$$\leq \frac{1}{5} + \frac{3}{20} + \frac{3}{20} = \frac{1}{2}. \tag{37}$$

Hence the decryption $\mathbf{Dec}(\mathbf{S}, \mathbf{y})$ outputs $\mathbf{m} = \mathbf{vH}$. We have $\mathbf{v} = \mathbf{mH}^{-1}$, since $\mathbf{H} = \mathsf{G}_{\mathrm{FRD}}(b) - \mathsf{G}_{\mathrm{FRD}}(b^*)$ is invertible for all $b \neq b^*$. The input vector $\mathbf{x} \in \{0,1\}^n$ can be recover correctly from the vector $\mathbf{v}$.

We now analyze the lossiness. For any input $\mathbf{x} \in \{0,1\}^n$, we have

$$\mathbf{y} = E_{\mathbf{S}}(\mathbf{0} = \mathbf{x}(\mathbf{0} \otimes \mathbf{g}), \mathbf{x}(\mathbf{U}+\mathbf{U}''), \mathbf{x}(\mathbf{U}'+\mathbf{U}'''); \mathbf{xA}, \mathbf{xE}). \quad (38)$$

For every $j \in [m]$, y_j is a ciphertext $(\mathbf{xA}, y'_j) \in \mathbb{Z}_q^l \times \mathbb{Z}_g$, where $\mathbf{xA}$ is the same randomness for all j and $y'_j = g(\langle \mathbf{xA}, \mathbf{s}_j \rangle + (\mathbf{xE})_j + 0 + 0)/q + (\mathbf{xU}')_j + (\mathbf{xU}''')_j)$. Fixing $\mathbf{A}$, $\mathbf{x}$ and $j \in [m]$, we have

$$|\frac{g}{q}(\mathbf{xE})_j + (\mathbf{xU}')_j + (\mathbf{xU}''')_j| \quad (39)$$

$$< \frac{g}{q} \cdot \frac{q}{5p} + n \leq \frac{q}{5p} + \frac{3q}{20p} < \frac{q}{2p}. \quad (40)$$

Obviously, the total number of outputs of the lossy function is at most $q^l(q/p)^m$, the logarithm of which gives an upper bound on the residual leakage

$$n - k \leq l \cdot \lg q + m \cdot \lg(\frac{q}{p}) \quad (41)$$

$$= n \cdot \frac{l \lg q}{m\lfloor \lg p \rfloor} + \frac{n}{\lfloor \lg p \rfloor} \cdot \lg(\frac{q}{p}) \quad (42)$$

$$\leq n \cdot \left(\frac{l}{m} \cdot \frac{\lg q}{\lg p - 1} + \frac{1}{\lg p - 1} \cdot \lg(\frac{q}{p}) \right) \quad (43)$$

$$= n \cdot \left(\frac{l}{m} \cdot \frac{\lg q}{\lg p - 1} + \log_{\frac{p}{2}}(\frac{q}{p}) \right) \quad (44)$$

$$= n \cdot \left(\frac{l}{m} \cdot \frac{\lg p}{\lg p - 1} + (\frac{l}{m} + 1) \log_{\frac{p}{2}}(\frac{q}{p}) \right). \quad (45)$$

Finally, note that the hidden lossy branch property follows from Lemma 2 on the security of matrix encryption. □

6. Parameter Instantiation and Worst-Case Connection

We now associate the security of our constructions with the worst-case quantum hardness of lattice problems [21, 25] in a black box manner, by properly instantiating all the parameters n, p, q, etc., and by invoking Proposition 1. The relationship between any desired constant lossiness rate $K \in (0,1)$, where larger K means more information is lost, and the corresponding approximation factor of the lattice problems is what we are interested in.

The following theorem is similar to the one in [8], except that the parameters we choose might not be the same as that of [8]. For completeness, the proof is presented here.

Theorem 4. For any constant $K \in (0,1)$, the construction of Section 4 with prime q gives a family of almost-always (n, Kn)-lossy TDF for all sufficiently large n, assuming that SIVP and GapSVP are hard for quantum algorithms to approximate to within $\tilde{O}(l^c)$ factors, where $c = 2 + \frac{3}{2(1-K)} + \delta$ for any desired $\delta > 0$.

The same applies for the construction in Section 5.2, with prime q and p, of almost-always (n, Kn)-all-but-one TDF.

In particular, by Proposition 1, the constructions are secure assuming that either SIVP or GapSVP are hard for quantum algorithms to approximate to within $\tilde{O}(l^{2+c})$ factors.

Proof. Using the notation from Theorem 1 (likewise Theorem 3), we let $p = n^{c_1}(p \in [n^{c_1}, (n+1)^{c_1}]$ is a prime) and let $n = l^{c_3}$ for some constant $c_1 > 1$, $c_3 > 1$ respectively that we will be set later, and let $r = n$, $\alpha = 1/(16pn)$. In order to invoke Proposition 1 (connecting LWE to lattice problems), we need to use some

$$q > 2\sqrt{l}/\alpha = 64pn\sqrt{l} = 64pn^{1+1/(2c_3)}. \quad (46)$$

Therefore we set $c_2 = 1 + 1/(2c_3)$, so we may take $q = O(pn^{c_2})$.

Now invoking Theorem 1 (likewise Theorem 3), we get that the lossy function has $n - k$ at most

$$n \cdot \left(\frac{c_2}{c_1} + \epsilon \right) = n \cdot \left(\frac{1 + 2c_3}{2c_1c_3} + \epsilon \right), \quad (47)$$

for any $\epsilon > 0$ and sufficiently large n. By Proposition 1, LWE is hard for our choice of parameters, assuming the lattice problems are hard to approximate within $\tilde{O}(l/\alpha) = \tilde{O}(l^{1+c_3(c_1+1)})$ factors for quantum algorithms. With the constraint on the residual leakage as $\frac{1+2c_3}{2c_1c_3} < 1 - K$, we get that $c_1 > \frac{1+2c_3}{2c_3(1-K)}$. This implies that the exponent in the lattice approximation factor may be brought arbitrarily close to $1 + c_3 + \frac{1+2c_3}{2(1-K)}$. Then under the constraint that $c_3 > 1$, the exponent may be brought arbitrarily close to $2 + \frac{3}{2(1-K)}$. □

7. Comparison

Let the parameters n, p, q, g, r, α be chosen as above. Compared to the LWE-based LTDF and ABO-TDF proposed in [8], our compact LTDF and ABO-TDF constructions reduce both the size of public key (i.e.the function index matrices) and that the vector of ciphertexts. Furthermore, the number of branches in our ABO-TDF is larger that of [8] if we fix p, which means that we can choose smaller p in order to support super-polynomially many injective branches in the construction of CCA secure public key encryption. The comparison is summarized in Table 1, where D-value stands for the corresponding difference value.

In [8], the construction of LWE-based LTDF yields public key $\mathbf{C} \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^{n \times m}$ and m-dimension vector of ciphertexts $\mathbf{y}$ where $y_j \in \mathbb{Z}_q^l \times \mathbb{Z}_q$. While the construction of LWE-based ABO-TDF in [8] yields public key $\mathbf{C} \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^{n \times w}$, where $w = m + 2l$, and w-dimension vector

Table 1. <Comparison between LTDF/ABO-TDF in [PW08] and those in this paper.>

	[PW08]	this paper	D-value
size of pk (LTDF)	$(nl+nm)\log q$	$nl\log q + nm\log g$	$nm\log(q/g)$
size of pk (ABO-TDF)	$(nl+nw)\log q$	$nl\log q + nm\log g$	$2nl\log q + nm\log(q/g)$
size of ciphertexts (LTDF)	$m(l+1)\log q$	$m(l\log q + \log g)$	$m\log(q/g)$
size of ciphertexts (ABO-TDF)	$w(l+1)\log q$	$m(l\log q + \log g)$	$(2l^2+2l)\log q + m\log(q/g)$
number of branches (ABO-TDF)	p^l	p^m	$p^m - p^l$

of ciphertexts $\mathbf{y}$ where $y_j \in \mathbb{Z}_q^l \times \mathbb{Z}_q$. The branch set is $B = \mathbb{Z}_p^l$.

In this paper, the construction of LTDF yields public key $\mathbf{C} \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_g^{n\times m}$ and m-dimension vector of ciphertexts $\mathbf{y} = \mathbf{xC}$ where $y_j \in \mathbb{Z}_q^l \times \mathbb{Z}_g$. While the construction of ABO-TDF yields public key $\mathbf{C} \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_g^{n\times m}$ and m-dimension vector of ciphertexts $\mathbf{y}$ where $y_j \in \mathbb{Z}_q^l \times \mathbb{Z}_g$. The branch set is $B = \mathbb{Z}_p^m$.

Take the size of public key as example, the difference value (D-value) between [PW08] and this paper is $nm\log(q/g)$ bits. When $p = n^{c_1}$, $q = O(pn^{c_2})$, $g = 4pn$ and $m = n/\lfloor \lg p \rfloor$, we have $nm\log(q/g) > n\log O(n^{c_2-1})$, which can substantially improve the performance. As for the number of branches, under the same choice of p, the difference value between ABO-TDF in [8] and that in this paper is $p^m - p^l$. When prime $p \in [n^{c_1}, (n+1)^{c_1}]$, $n = l^{c_3}$ for some constants $c_1 > 1$ and $c_3 > 1$, we have $m = \frac{n}{\lfloor \log p \rfloor} \approx \frac{l^{c_3}}{c_3 c_1 \log l} = l \cdot \frac{l^{c_3-1}}{c_3 c_1 \log l} > l$, which means more branches under the same p. Alternatively, we only need smaller p (in turn smaller l since $p \in [l^{c_3 c_1}, (l^{c_3}+1)^{c_1}]$) to support enough branches.

8. The Applications of LTDF and ABO-TDF

In this section, for the completeness of this paper, we show that LTDF and ABO-TDF can be used for black-box constructions of many primitives and cryptosystems (including those in Section 2.1), by summarizing the facts in [8] without proofs. It needs to be emphasized that the performance of primitives and cryptosystems constructed from LTDF and ABO-TDF in this paper is much better that in [8] according to the results in Section 7.

8.1. Injective Trapdoor Functions and Hard-Core Functions

Lemma 1 says that we can construct injective trapdoor functions from LTDF in the standard sense, that is, easy to invert with a trapdoor but hard to invert otherwise.

A family of functions $\mathcal{H} = \{h : D \to R\}$ from a domain D to range R is said to be *universal* if, for every distinct $x, x' \in D$, $\Pr_{h\leftarrow\mathcal{H}}[h(x) = h(x')] = 1/|R|$. Let $(S_{\mathrm{ltdf}}, F_{\mathrm{ltdf}}, F_{\mathrm{ltdf}}^{-1})$ give a collection of (n,k)-LTDF. Let $\mathcal{H}$ be an universal family of hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$, where $\ell \le k - 2\lg(1/\epsilon)$ for some negligible $\epsilon = \mathrm{negl}(\lambda)$. Define the following distributions:

- X_0: choose $(s,t) \leftarrow S_{\mathrm{inj}}$, $h \leftarrow \mathcal{H}$, and $x \leftarrow \{0,1\}^n$. Output $(s, h, F_{\mathrm{ltdf}}(s,x), h(x))$.
- X_1: choose $(s,t) \leftarrow S_{\mathrm{loss}}$, $h \leftarrow \mathcal{H}$, and $x \leftarrow \{0,1\}^n$. Output $(s, h, F_{\mathrm{ltdf}}(s,x), h(x))$.
- X_2: choose $(s,t) \leftarrow S_{\mathrm{loss}}$, $h \leftarrow \mathcal{H}$, $x \leftarrow \{0,1\}^n$, and $r \leftarrow \{0,1\}^\ell$. Output $(s, h, F_{\mathrm{ltdf}}(s,x), r)$.
- X_3: choose $(s,t) \leftarrow S_{\mathrm{inj}}$, $h \leftarrow \mathcal{H}$, $x \leftarrow \{0,1\}^n$, and $r \leftarrow \{0,1\}^\ell$. Output $(s, h, F_{\mathrm{ltdf}}(s,x), r)$.

Lemma 4. Let X_0, X_1, X_2, X_3 be as defined above, then

$$\{X_0\} \overset{s}{\approx} \{X_1\} \overset{s}{\approx} \{X_2\} \overset{s}{\approx} \{X_3\}. \tag{48}$$

In particular, $\mathcal{H}$ is a family of hard-core functions for the lossy collection.

8.2. Universal One-Way and Collision Resistant Hashing

Note that a collection of CRHFs is also a collection of UOWHFs. Assume that the input length $n(\lambda) = \lambda$ equals the security parameter. Let $(S_{\mathrm{ltdf}}, F_{\mathrm{ltdf}}, F_{\mathrm{ltdf}}^{-1})$ give a collection of (n,k)-LTDF $\{f_s : \{0,1\}^n \to \mathcal{R}\}$ having arbitrary range $\mathcal{R}$ and residual leakage $n - k \le n/2 - d$ for some $d = \omega(\log\lambda)$. Let $\mathcal{H} = \{h : \mathcal{R} \to \{0,1\}^\ell\}$ be an universal family of hash functions, where $n - d \le \ell < n$.

We describe the algorithms for the collection of CRHFs as follows:

- S_{crh} chooses $(s,t) \leftarrow S_{\mathrm{inj}}$ and disposes of t. It also chooses $h \leftarrow \mathcal{H}$. The index of the hash function is $i = (s,h)$.
- $F_{\mathrm{crh}}(i,x)$ on input index $i = (s,h)$ and $x \in \{0,1\}^n$, outputs $h(F_{\mathrm{ltdf}}(s,x)) \in \{0,1\}^\ell$.

Lemma 5. The algorithms $(S_{\mathrm{crh}}, F_{\mathrm{crh}})$ described above give a collection of collision-resistant hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$.

8.3. Cryptosystems and Oblivious Transfer

In fact, CPA-secure cryptosystem implies oblivious transfer and multiparty computation protocol. We omit the details here and only describe the results of CPA- and CCA-secure construction. Reader may refer to [29] for more information.

CPA-Secure Construction. Let $(S_{\text{ltdf}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$ give a collection of (n,k)-LTDF (or almost always LTDF). Let $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^\ell\}$ be an universal family of hash functions, where $\ell \le k - 2\log(1/\epsilon)$ for some $\epsilon = \text{negl}(\lambda)$. The message space is $\{0,1\}^\ell$ and the construction is as follows.

- **Key generation.** $\mathcal{G}$ generates an injective trapdoor function as $(s,t) \leftarrow S_{\text{inj}}$, and chooses a hash function $h \leftarrow \mathcal{H}$. The public key $pk = (s,h)$, and the secret key $sk = (t,h)$.
- **Encryption.** $\mathcal{E}$ on input a public key $pk = (s,h)$ and a message $m \in \{0,1\}^\ell$. It chooses $x \leftarrow \{0,1\}^n$ uniformly at random. The ciphertext is $c = (c_1, c_2)$, where

$$c_1 = F_{\text{ltdf}}(s,x), \quad c_2 = m \oplus h(x). \tag{49}$$

- **Decryption.** $\mathcal{D}$ on input a secret key $sk = (t,h)$ and a ciphertext $c = (c_1, c_2)$. It computes $x = F^{-1}_{\text{ltdf}}(t, c_1)$ and outputs $c_2 \oplus h(x)$.

Theorem 5. The algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ described above give a CPA-secure cryptosystem.

CCA-Secure Construction. First, let $(\text{Gen}, \text{Sign}, \text{Ver})$ be a strongly unforgeable one-time signature scheme, where the public verification keys are in $\{0,1\}^\nu$. Let $(S_{\text{ltdf}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$ give a collection of (n,k)-LTDF (or almost always LTDF). Let $(S_{\text{abo}}, G_{\text{abo}}, G^{-1}_{\text{abo}})$ give a collection of (n,k')-ABO trapdoor functions (or almost always ABO-TDF) with the branch set $B = \{0,1\}^\nu$, which contains the set of signature verification keys. Let $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^\ell\}$ be an universal family of hash functions, where $0 < \ell \le k - 2\log(1/\epsilon)$ for some $\epsilon = \text{negl}(\lambda)$.

Next, we require that the total residual leakage of the lossy and ABO collections is

$$r + r' = (n-k) + (n-k') \le n - \kappa, \tag{50}$$

where $\kappa = \kappa(n) = \omega(\log n)$. The message space is $\{0,1\}^\ell$ and the construction is as follows.

- **Key generation.** $\mathcal{G}$ generates an injective trapdoor function as $(s,t) \leftarrow S_{\text{inj}}$, an ABO-TDF having lossy branch 0^ν via $(s',t') \leftarrow S_{\text{abo}}(0^\nu)$, and chooses a hash function $h \leftarrow \mathcal{H}$. The public key $pk = (s,s',h)$, and the secret key $sk = (t,t',pk)$.
- **Encryption.** $\mathcal{E}$ on input a public key $pk = (s,s',h)$ and a message $m \in \{0,1\}^\ell$. It generates one-time signature keypair $(vk, sk_\sigma) \leftarrow \text{Gen}$, then chooses $x \leftarrow \{0,1\}^n$ uniformly at random. Next it computes

$$c_1 = F_{\text{ltdf}}(s,x), c_2 = G_{\text{abo}}(s', vk, x), c_3 = m \oplus h(x). \tag{51}$$

  Finally, it signs the above tuple (c_1, c_2, c_3) as $\sigma \leftarrow \text{Sign}(sk_\sigma, (c_1, c_2, c_3))$.
  The output ciphertext is

$$c = (vk, c_1, c_2, c_3, \sigma). \tag{52}$$

- **Decryption.** algorithm $\mathcal{D}$ on input a secret key $sk = (t, t', pk = (s, s', h))$ and a ciphertext $c = (vk, c_1, c_2, c_3, \sigma)$. First, it checks that whether $\text{Ver}(vk, (c_1, c_2, c_3), \sigma) = 1$; if not, it outputs $\perp$. It then computes $x = F^{-1}_{\text{ltdf}}(t, c_1)$ and checks that whether $c_1 = F_{\text{ltdf}}(s,x)$ and $c_2 = G_{\text{abo}}(s', vk, x)$; if not, it outputs $\perp$. Finally, it outputs $m = c_3 \oplus h(x)$.

Theorem 6. The algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ described above give a CCA-secure cryptosystem.

References

[1] Naor, M. and Yung, M. (1990) Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90 (New York, NY, USA: ACM): 427–437. doi:10.1145/100216.100273, URL http://doi.acm.org/10.1145/100216.100273.

[2] Rackoff, C. and Simon, D.R. (1992) Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91 (London, UK, UK: Springer-Verlag): 433–444. URL http://dl.acm.org/citation.cfm?id=646756.705376.

[3] Dolev, D., Dwork, C. and Naor, M. (2000) Nonmalleable cryptography. *SIAM J. Comput.* **30**(2): 391–437. doi:10.1137/S0097539795291562, URL https://doi.org/10.1137/S0097539795291562.

[4] Diffie, W. and Hellman, M.E. (1976) New directions in cryptography. *IEEE Transactions on Information Theory* 22(6): 644–654.

[5] Rivest, R.L., Shamir, A. and Adleman, L. (1978) A method for obtaining digital signatures and public-key cryptosystems. In *Communications of the ACM*: 120–126.

[6] Rabin, M.O. (1979) *DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS INTRACTABLE AS FACTORIZATION* (Massachusetts Institute of Technology).

[7] Paillier, P. (1999) Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on Theory and Application of Cryptographic Techniques*: 223–238.

[8] Peikert, C. and Waters, B. (2008) Lossy trapdoor functions and their applications. In *Fortieth ACM Symposium on Theory of Computing*: 187–196.

[9] Sahai, A. (1999) Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99 (Washington, DC, USA: IEEE Computer Society): 543–553. URL http://dl.acm.org/citation.cfm?id=795665.796535.

[10] Cramer, R. and Shoup, V. (1998) A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '98 (London, UK, UK: Springer-Verlag): 13–25. URL http://dl.acm.org/citation.cfm?id=646763.706340.

[11] Cramer, R. and Shoup, V. (2002) Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02 (London, UK, UK: Springer-Verlag): 45–64. URL http://dl.acm.org/citation.cfm?id=647087.715842.

[12] Goldreich, O. (2001) *The Foundations of Cryptography - Volume 1, Basic Techniques* (DBLP).

[13] Goldreich, O. (2004) *Foundations of Cryptography: Volume 2, Basic Applications* (Cambridge University Press).

[14] Rackoff, C. and Simon, D.R. (1991) Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *International Cryptology Conference on Advances in Cryptology*: 433–444.

[15] Boldyreva, A., Fehr, S. and O'Neill, A. (2008) On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Conference on Cryptology: Advances in Cryptology*: 335–359.

[16] Kiltz, E. and OąŕNeill, A. (2010) *Instantiability of RSA-OAEP under Chosen-Plaintext Attack* (Springer Berlin Heidelberg).

[17] Hofheinz, D. (2011) Possibility and impossibility results for selective decommitments. *Journal of Cryptology* **24**(3): 470–516.

[18] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A. and Segev, G. (2010) More constructions of lossy and correlation-secure trapdoor functions. In *International Workshop on Public Key Cryptography*: 279–295.

[19] Xue, H., Li, B., Lu, X., Jia, D. and Liu, Y. (2013) *Efficient lossy trapdoor functions based on subgroup membership assumptions.* (Berlin: Springer), 235–250. doi:10.1007/978-3-319-02937-5_13.

[20] Seurin, Y. (2014) *On the Lossiness of the Rabin Trapdoor Function* (Springer Berlin Heidelberg).

[21] Regev, O. (2005) On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05 (New York, NY, USA: ACM): 84–93. doi:10.1145/1060590.1060603, URL http://doi.acm.org/10.1145/1060590.1060603.

[22] Blum, A., Kalai, A. and Wasserman, H. (2000) Noise-tolerant learning, the parity problem, and the statistical query model. *CoRR* **cs.LG/0010022**. URL http://arxiv.org/abs/cs.LG/0010022.

[23] Cramer, R. and Damgård, I. (2009) On the amortized complexity of zero-knowledge protocols. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*: 177–191. doi:10.1007/978-3-642-03356-8_11, URL https://doi.org/10.1007/978-3-642-03356-8_11.

[24] Applebaum, B., Cash, D., Peikert, C. and Sahai, A. (2009) Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009, International Cryptology Conference, Santa Barbara, Ca, Usa, August 16-20, 2009. Proceedings*: 595–618.

[25] Peikert, C. (2009) Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *ACM Symposium on Theory of Computing*: 333–342.

[26] Ajtai, M., Kumar, R. and Sivakumar, D. (2001) A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01 (New York, NY, USA: ACM): 601–610. doi:10.1145/380752.380857, URL http://doi.acm.org/10.1145/380752.380857.

[27] Lenstra, H.W., Lenstra, A.K. and Lovfiasz, L. (1982) Factoring polynomials with rational coeficients: 515–534.

[28] Schnorr, C. (1987) A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.* **53**: 201–224. doi:10.1016/0304-3975(87)90064-8, URL https://doi.org/10.1016/0304-3975(87)90064-8.

[29] Peikert, C. and Waters, B. (2011) Lossy trapdoor functions and their applications. *SIAM J. Comput.* **40**(6): 1803–1844. doi:10.1137/080733954, URL https://doi.org/10.1137/080733954.

[30] Peikert, C. (2007) Limits on the hardness of lattice problems in ℓ_p norms. In *IEEE Conference on Computational Complexity*: 333–346.

[31] Agrawal, S., Dan, B. and Boyen, X. (2010) *Efficient Lattice (H)IBE in the Standard Model* (DBLP).