Cluster Computing (2019) 22:S5647–S5654
https://doi.org/10.1007/s10586-017-1444-9

Lossy trapdoor functions based on the PLWE

Chengli Zhang[1] · Wenping Ma[1] · Hefeng Chen[2] · Feifei Zhao[1]

Received: 27 July 2017 / Revised: 20 November 2017 / Accepted: 23 November 2017 / Published online: 12 December 2017

Abstract
In 2011, Chris Peikert and Brent Waters proposed the concept of lossy trapdoor functions, which is an inherent and powerful cryptographic concept. Lossy trapdoor functions can be used for simple black-box constructing CCA encryption schemes, collision-resistent hash functions and oblivious transfer schemes. Chris Peikert and Brent Waters constructed lossy trapdoor functions based on decisional Diffie–Hellman assumption and learning with errors problem separately, which can be generalized to all-but-one trapdoor functions. In this paper, we generalize the lossy trapdoor functions and all-but-one trapdoor functions based on the polynomial ring separately, and we construct two types of trapdoor functions based on polynomial learning with errors assumption, which have more throughput and efficiency.

Keywords Lattices · Lossy trapdoor functions · All-but-one trapdoor functions · Polynomial learning with errors

1 Introduction

A central goal in cryptography is to realize a variety of security notions based on plausible and concrete computational assumptions. The assumptions have typically been concerned with problems from three categories: factoring large integers [1–5], computing discrete logarithms in cyclic groups [6–8], computational problems on lattices [9–13]. In public key cryptography, two important notions are trapdoor functions (TDFs) and security under chosen ciphertext attack (CCA security). Trapdoor functions were first realized by the RSA function of Rivest, Shamie, and Adleman. While CCA security has become the notion of security for public key encryption under active attacks.

In resent years, lattices have raised as a very attractive foundation for cryptography [14–18]. The appeal of lattice-based primitives stems from the fact that the security can often be based on worst-case hardness assumptions. Much more recent research in lattice cryptography focus on ring-based primitives such as ring-LWE [19,20].

A lattice has a typical linear structure and some computational problems about it have been proven to be NP-hard. Many exciting developments in lattice-based cryptography have occurred in the past few years, and there have been renewed interest in lattice-based cryptography as prospects for a real quantum computer improve. As is well known, some lattice-based cryptosystems can be resistant to attack by both classical and quantum computers.

Before the year of 2008, for CCA security, the main approach in the existing literature relies on non-interactive zero-knowledge (NIZK) proofs. Cryptosystems have been constructed based on problems related to factoring and discrete logs, but not lattices. For trapdoor functions, the state of the art is even less satisfactory: though TDFs are widely viewed as a general primitive, they have so far been realized only from problems related to factoring.

In 2011, Chris Peikert and Brent Waters proposed the definition of lossy trapdoor functions based on lattice whose existence implies that of general trapdoor functions [13]. And it can be used to develop new approaches for injective trapdoor functions, constructing collision resistant hash functions, oblivious transfer sch-emes, and chosen ciphertext-secure cryptosystems. All of the above constructions are simple, efficient, and black-box. Chris Peikert and Brent Waters realized lossy TDFs under a variety of different number-theoretic assumptions, including hardness of the decisional Diffie–Hellman (DDH) problem, and the worst-case hardness of standard lattice problem for quantum algorithms (alternately, under an average-case hardness assumption for classical algorithms). These constructions are simple and

✉ Chengli Zhang
zcl0719@163.com

1 State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, People's Republic of China

2 Computer Engineering College, Jimei University, Xiamen 361021, People's Republic of China

excellent. Unfortunately, their public function size are too large. In special, when having n bits input, their description of the public key is $\Theta((n/logn)^2)$ bits. The new lossy trapdoor functions based on Demgard Jurik Encryption were constructed by Alon Rosen and Gil Segev [21]. Such construction have a short description of the public key, but the complexity is very large. In 2012, lossy trapdoor functions based on rounding technique and LWR assumption were constructed by Joel Alwen, which is simple [22]. But considered the problem of LWR, large parameters are needed and the complexity is also high.

In this paper, based on extending the definition of general lossy trapdoor functions, we proposed a new method of constructing lossy trapdoor functions based on ring polynomial LWE, and comparing with the previous lattice-based constructions of lossy trapdoor function, our scheme has the following advantages:

1. The larger throughput;
2. Higher velocity;
3. Extending the definition of lossy trapdoor functions which have the wider domain and the larger applied range;
4. Smaller function index;
5. Resisting quantum attack.

We now describe the our basic framework for constructing our new lossy trapdoor functions and all-but-one trapdoor functions.Firstly, we construct the basic encryption algorithm which has the property of additive homomorphism using the PLWE assumption (high efficiency because of no using rounding algorithm). Secondly, the basic encryption algorithm is extended to the matrix. For every element in the matrix, calling the encryption algorithm by choosing the same random element for the same row and choosing the same secret key for the same column. The independence of encryptions on elements of matrix enables the additive homomorphism property of the basic encryption algorithm to be passed to the matrix encryption algorithm. The advantage of cross -choice of the random elements and the secret keys is that greatly reducing the average number of random elements and secret keys on a single plaintext matrix in the process of matrix encryption. Thirdly, constructing our lossy trapdoor functions using matrix encryption algorithm. We also give the proof of the main theorem for the lossy trapdoor functions. And lastly, we give the advantage of our scheme.

2 Preliminaries

2.1 Notations

Let D be a distribution on a finite set S, $d \leftarrow D$ denotes choosing element by distribution D, and $d \leftarrow S$ denotes uniformly choosing element from S. $\mathbb{Z}[x]$ is polynomial ring on integer ring. Let $f(x) \in \mathbb{Z}[x]$, then the maximum degree of $f(x)$ is written as $\partial^0 f(x) = n$ and $\mathbb{Z}[x]/(f(x))$ denotes the residue ring modular $f(x)$. $\mathbb{Z}_q$ denotes the ring modular a integer q, $\mathbb{Z}_q[x]$ denotes the polynomial integer ring on $\mathbb{Z}_q$, and $\mathbb{Z}_q[x]/(f(x))$ is the residue ring modular $f(x)$ on $\mathbb{Z}_q[x]$.

Let $\mathbf{v}$ be a vector, then $\|\mathbf{v}\|$ denotes $l_\infty-$ norm of a vector $\mathbf{v}$, that is, $\|v\| = max_i \mid v_i \mid$. The norm $\| \ p(x) \ \|$ of a polynomial $p(x)$ denotes the norm of its coefficient vector.

Let $\mathbf{A}$ be an $n \times m$ matrix and $\mathbf{B}$ be an $n \times m'$ matrix, then $(\mathbf{A}\,\mathbf{B})$ denotes the $n \times (m+m')$ matrix formed by concatenating $\mathbf{A}$ and $\mathbf{B}$. Similarly, suppose that $\mathbf{A}$ has dimensions $n \times m$ and $\mathbf{B}$ is an $n' \times m$, then $\begin{pmatrix}\mathbf{A}\\ \mathbf{B}\end{pmatrix}$ is the $(n+n') \times m$ matrix formed by putting $\mathbf{A}$ on top of $\mathbf{B}$. Similarly, the notations apply to vectors. When doing matrix-vector multiplication we usually view vectors as column vectors.

Each $n-1$-th degree polynomial on $\mathbb{Z}$ corresponds with an n-dimensional vector on $\mathbb{Z}$. An injective function from the set of all the $n-1$-th degree polynomial on $\mathbb{Z}$ to the set $\mathbb{Z}^n$ has been constructed. Let $\mathbf{p}$ be the coefficient vector of the polynomial $p(x)$.

Let $D_{\mathbb{Z}^n,r}$ be an n-dimensional discrete Guassion distribution, then $D_{\mathbb{Z}^n,r}$ becomes the distribution of the $n-1$-th degree polynomials on $\mathbb{Z}$.

2.2 Lattice [9]

We will review some basic definitions about lattice. A lattice in R^n is defined as the set of all integer combinations of n linearly independent vectors. This set of vectors is known as a basis of the lattice and the basis is not unique.

Definition 2.1 An n-dimensional lattice is the set of all integer combinations

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b_i} : x_i \in \mathbb{Z} \ for \ 1 \le i \le n \right\}$$

of n linearly independent vectors $\mathbf{b_1}, \cdots, \mathbf{b_n}$ in $\mathbb{R}^n$.

The set of vectors $\mathbf{b_1}, \cdots, \mathbf{b_n}$ is called a basis for the lattice. A basis can be represented by the matrix $\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_n}) \in \mathbb{R}^{n \times n}$ where the basis vectors are columns of matrix. The basis of a lattice is not unique and there is only a difference of one unimodular matrix between the bases. Usually, we want to find the relatively short basis. $\mathcal{L}(\mathbf{B})$ denotes the lattice generated by $\mathbf{B}$. Notice that $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication.

The dual of a lattice Λ in R^n, denoted $\Lambda^\vee$, is the lattice given by the set of all vectors $\mathbf{y} \in R^n$ such that $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ for all vectors $\mathbf{x} \in \Lambda$.

2.3 Trapdoor functions [13]

We recall one standard definition of a collection of injective trapdoor functions (TDFs).

Definition 2.2 For generality, let $n = n(\lambda) = poly(\lambda)$ denote the input length of the trapdoor functions as a function of the security parameter λ. A collection of injective trapdoor functions is given 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 (deterministic) 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 $\mathcal{I}$, the probability that $\mathcal{I}(s, f_s(x))$ outputs x is negligible, where the probability is taken over the choice of $(s, t) \leftarrow S, x \leftarrow \{0, 1\}^n$, and $\mathcal{I}$ is randomness.

2.4 Lossy trapdoor functions [13]

Lossy trapdoor functions (lossy TDFs) is a general cryptographic primitive.

Definition 2.3 Let $n(\lambda) = poly(\lambda)$ represent the input length of the function and $k(\lambda) \leq n(\lambda)$ represent the lossiness of the collection. For convenience, define the residual leakage as $r(\lambda) := n(\lambda) - k(\lambda)$. For all the above quantities, the dependence on λ is omitted.

A collection of (n, k)-lossy trapdoor functions is given by a tuple of PPT algorithms $(S_{ltsf}, F_{ltdf}, F_{ltdf}^{-1})$ having the properties below. For notational convenience, define the algorithms $S_{inj}(\cdot) := S_{ltdf}(\cdot, 1)$ and $S_{loss}(\cdot) := S_{ltdf}(\cdot, 0)$.

1. Easy to sample an injective function with trapdoor: S_{inj} outputs (s, t) where s is a function index and t is its trapdoor, $F_{ltdf}(s, \cdot)$ computes an injective (deterministic) function $f_s(\cdot)$ over the domain $\{0, 1\}^n$, and $F_{ltdf}^{-1}(t, \cdot)$ computes $f_s^{-1}(\cdot)$. If a value y is not in the image of f_s, i.e., if $f_s^{-1}(y)$ does not exist, then the behavior of $F_{ltdf}^{-1}(t, y)$ is unspecified (because of this, the output of F_{ltdf}^{-1} may need to be checked for correctness in certain applications).
2. Easy to sample a lossy function: S_{loss} outputs $(s, \perp)$ where s is a function index, and $F_{ltdf}(s, \cdot)$ computes a (deterministic) function $f_s(\cdot)$ over the domain $\{0, 1\}^n$ whose image has size at most $2^r = 2^{n-k}$.
3. Hard to distinguish injective from lossy: the first outputs of S_{inj} and S_{loss} are computationally indistinguishable. More formally, let X_λ denote the distribution of s from S_{inj}, and let Y_λ denote the distribution of s from S_{loss}. Then $X_\lambda \approx Y_\lambda$.

The property that an injective function is hard to invert is implied by combination of the lossiness and indistinguishability properties.

For many lattice-based constructions, a slightly relaxed definition of lossy trapdoor functions is considered, called almost-always lossy TDFs. Namely, it is required that with overwhelming probability over the randomness of S_{inj}, the index s of S_{inj} describes an injective function f_s that F_{ltdf}^{-1} inverts correctly on all values in the image of f_s. In other words, there is a negligible probability over the choice of s that $f_s(\cdot)$ is not injective or that $F_{ltdf}^{-1}(t, \cdot)$ incorrectly computes $f_s^{-1}(\cdot)$ for some input. And the use of almost-always lossy TDFs does not affect the security in some applications.

2.5 All-but-one trapdoor functions [13]

In all-but-one (ABO) trapdoor functions, each function has an extra input called its branch. All of the branches are injective trapdoor functions (having the same trapdoor value), expect for one branch which is lossy. The lossy branch is specified as a parameter to the function sampler, and the value is hidden by the resulting function description.

Definition 2.4 The parameters n, k, r are the same as in Definition 2.3, and also let $\mathcal{B} = \{B_\lambda\}_{\lambda \in N}$ be a collection of sets whose elements represent the branches. Then a collection of $(n, k) - all - but - one$ trapdoor functions with branch collection $\mathcal{B}$ is given by a tuple of PPT algorithms $(S_{abo}, G_{abo}, G_{abo}^{-1})$ having the following properties:

1. Sampling a trapdoor function with given lossy branch: for any $b^* \in B_\lambda$, $S_{abo}(b^*)$ outputs (s, t), where s is a function index and t is its trapdoor. For any $b \in B_\lambda$ distinct from b^*, $G_{abo}(s, b, \cdot)$ computes an injective (deterministic) function $g_{s,b}(\cdot)$ over the domain $\{0, 1\}^n$, and $G_{abo}^{-1}(t, b, \cdot)$ computes $g_{s,b}^{-1}(\cdot)$. As above, the behavior of $G_{abo}^{-1}(t, b, y)$ is unspecified if $g_{s,b}^{-1}(y)$ does not exist. Additionally, $G_{abo}(s, b^*, \cdot)$ computes a function $g_{s,b^*}(\cdot)$ over the domain $\{0, 1\}^n$ whose image has size at most $2^r = 2^{n-k}$.
2. Hidden lossy branch: for any $b_0^*, b_1^* \in B_\lambda$, the first output s_0 of $S_{abo}(b_0^*)$ and the first output s_1 of $S_{abo}(b_1^*)$ are computationally indistinguishable.

An almost-always relaxation of the ABO trapdoor functions definition can be also taken into account. Specifically, the injective, invertible, lossy properties need only hold with overwhelming probability over the choice of the function index s. Similarly, the use of the almost-always ABO collection does not affect security in many applications.

2.6 The learning with errors problem [23]

The “Learning with errors” (LWE) problem is to distinguish random linear equations, which have been perturbed by a small amount of noise, from truly uniform ones. LWE problem has been showed to be as hard as worse-case lattice problem and it has served as the foundation for many cryptography applications.

Definition 2.5 Fix a positive integer n, integers $m \geq n$ and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on the interval $[0, q)^m$. Define the following two distributions over $\mathbb{Z}_q^{n\times m} \times [0, q)^m$:

1. $LWE_{m,q,\chi}(\mathbf{s})$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, sampling $\mathbf{e} \leftarrow \chi$, and outputting $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}\ mod\ q)$.
2. $U_{m,q}$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and uniform $\mathbf{y} \in [0, q)^m$, and outputting $(\mathbf{A}, \mathbf{y})$.

The decisional variant of the LWE problem can be stated informally as the problem of distinguishing between $U_{m,q}$ and $LWE_{m,q,\chi}(\mathbf{s})$ for a uniform $\mathbf{s}$. Formally, for m, q, and χ that may depend on n (viewed now as a security parameter), we say the $LWE_{m,q,\chi}$ problem is hard if the following is negligible for any probabilistic polynomial-time algorithm D:

$$|Pr[\mathbf{s} \leftarrow \mathbb{Z}_q^n;\ (\mathbf{A}, \mathbf{y}) \leftarrow LWE_{m,q,\chi}(\mathbf{s}) : D(\mathbf{A}, \mathbf{y}) = 1] \\ -Pr[(\mathbf{A}, \mathbf{y}) \leftarrow U_{m,q} : D(\mathbf{A}, \mathbf{y}) = 1]|.$$

The polynomial learning with errors (PLWE) assumption is analogous to the learning with errors assumption. In the PLWE assumption, we consider the rings $R = \mathbb{Z}[x]/\langle f(x)\rangle$ and $R_q = R/qR$ for some degree n integer polynomial $f(x) \in \mathbb{Z}[x]$ and a prime integer $q \in \mathbb{Z}$. And $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$, that is, the ring of degree $n-1$ polynomials with coefficients in $\mathbb{Z}_q$. Addition in the rings is done component-wise in their coefficients and multiplication is simply polynomial multiplication modular $f(x)$. The same as q, in the case of the ring R_q.

The element in R can be viewed as a degree $n-1$ polynomial over $\mathbb{Z}$. And the $PLWE_{f,q,\chi}$ assumption is parameterized by the integer polynomial $f(x) \in \mathbb{Z}[x]$ of degree n, the prime integer $q \in \mathbb{Z}$, the error distribution χ over R.

Definition 2.6 Let λ be a security parameter, let $f(x) = f_\lambda(x) \in \mathbb{Z}[x]$ be a polynomial of degree $n = n(\lambda)$, let $q = q(\lambda) \in \mathbb{Z}$ be a prime integer, let the ring $R = \mathbb{Z}[x]/\langle f(x)\rangle)$ and $R_q = R/qR$, and let χ denote a distribution over the the ring R.

The polynomial LWE assumption $PLWE_{f,q,\chi}$ states that for any $l = poly(\lambda)$, $\{(a_i, a_i \cdot z + e_i)\}_{i\in[l]}$ and $\{(a_i, u_i)\}_{i\in[l]}$ are computationally indistinguishable where z is sampled from the noise distribution χ, a_i are uniform in R_q, the“error polynomials” e_i are sampled from the error distribution χ, and finally, the ring elements u_i are uniformly random over R_q.

When the indistinguishability is required to hold given only l samples (for some $l = poly(\lambda)$), the assumption is denoted by $PLWE^{(l)}_{f,q,\chi}$.

Note that the definition of the $PLWE$ assumption is defined as a decisional assumption. While the search assumption is defined which requires an adversary to find $s \in R_q$, and any polynomial number of samples $(a_i, a_i \cdot s + e_i)$. For some range of parameters, the search and decisional assumptions are equivalent.

Proposition 2.7 *Let $f(x)$, q and χ be as in Definition 27. Let $t = t(\lambda) \in \mathbb{Z}_q^*$ (thus t and q are relatively prime). Then for any $l = poly(\lambda)$, the $PLWE^{(l)}_{f,q,\chi}$ assumption implies that,*

$$\{(a_i, a_i \cdot z + te_i)\}_{i\in[l]} \approx_c \{(a_i, u_i)\}_{i\in[l]},$$

where a_i, z, e_i and u_i are as in Definition 2.6.

3 New lossy trapdoor functions

Let λ be a security parameter. Let $q = q(\lambda)$ be a prime integer, and $t = t(\lambda) \in \mathbb{Z}_q^*$ be a prime integer. Let $f(x) \in \mathbb{Z}[x]$ be an n-th degree polynomial. Let $R_q = \mathbb{Z}_q[x]/(f(x))$. Let $D_{\mathbb{Z}^n,r}$ be n-dimensional discrete Gaussian distribution where $r > 0$. Let χ be a distribution on ring $R_q = \mathbb{Z}_q[x]/(f(x))$. In this scheme, n, f, q, χ are public parameters.

Zvika Brakerski and Vinod Vaikuntanathan construct a full homomorphism encryption scheme based on Hermite Normal polynomial LWE assumption [23]. In this paper, we construct a matrix encryption scheme based on general Hermite Normal polynomial LWE assumption and then on this basis lossy trapdoor functions and all-but-one trapdoor functions are constructed. The message space in this paper is R_t, that is, coded messages is n-th polynomial on $\mathbb{Z}_t$. We suppose that $t^2 \cdot r \cdot n^{1.5} \leq q/2$ for correctly encryption.

We now describe the our basic framework for constructing our new lossy trapdoor functions and all-but-one trapdoor functions.

Firstly, we construct the basic encryption algorithm which has the property of additive homomorphism using the PLWE assumption (high efficiency because of no using rounding algorithm). The algorithm obtain the additive homomorphic ciphertext $c = (a, az + te + m)$ by adding a multiplication az of a random element a and the key z and t times the error

e to the plaintext m on R_t. Such construction can make the decryption algorithm simple and easy to operate that is only needed to subtract the first term of the ciphertext from the second term and then take modular t operation.

Secondly, the basic encryption algorithm is extended to the matrix. For every element in the matrix, calling the basic encryption algorithm by choosing the same random element for the same row and choosing the same secret key for the same column. The independence of encryptions on elements of matrix enables the additive homomorphism property of the basic encryption algorithm to be passed to the matrix encryption algorithm. The advantage of cross -choice of the random elements and the secret keys is that greatly reducing the average number of random elements and secret keys on a single plaintext matrix in the process of matrix encryption. Encrypting a matrix $\mathbf{M} \in R_t^{h\times h}$, that is, encrypting h^2 elements in the matrix, needs h random elements and h secret keys instead of h^2 random elements and h^2 secret keys. In addition, the additive homomorphic property of the basic encryption algorithm guarantees the linearity of the matrix encryption algorithm, that is, if we obtain the ciphertext matrix $\mathbf{C}$ by encrypting the plaintext $\mathbf{M}$, for the vector $\mathbf{x}$, we have that

$$\mathbf{xC} = E_{\mathbf{z}}(\mathbf{xM}; \mathbf{xa}, \mathbf{xE}),$$
$$\mathbf{C} + \mathbf{M}' = E_{\mathbf{z}}(\mathbf{M} + \mathbf{M}'; \mathbf{a}, \mathbf{E}).$$

Note that, when $\mathbf{C}$ is the encryption of $\mathbf{M}$, then we can compute that $\mathbf{C} + \mathbf{M}'$ is the encryption of $\mathbf{M} + \mathbf{M}'$ without the secret information.

Thirdly, our lossy trapdoor functions will be constructed by using matrix encryption algorithm. We also give the proof of the main theorem for the lossy trapdoor functions. In addition, we can also get the construction of our all-but-one trapdoor functions as the special case. Take all-but-one trapdoor functions as an example to describe our procedure. For each branch b, construct a diagonal branch matrix $\mathbf{M}$ whose every diagonal element is b. For the given lossy branch b^*, we encrypt $\mathbf{M}^*$ using matrix encryption algorithm, then the ciphertext $\mathbf{C}^*$ we get is the function index and the secret key is the trapdoor. In the G_{ABO} algorithm, we construct the function $g_{\mathbf{C}^*,b}$ whose function description is $\mathbf{C}^*$, the output of S_{ABO} algorithm, the branch is b, the input is $\mathbf{x}$, and the output is $\mathbf{y} = \mathbf{x}(\mathbf{C} + (\mathbf{0}\ \mathbf{b}))$, that is the encryption of $\mathbf{x}(-\mathbf{M}^* + (\mathbf{0}\ \mathbf{b}))$. Thus in the G_{ABO}^{-1} algorithm, the matrix decryption algorithm can be used to solve inversion with the trapdoor information.

3.1 Basic encryption algorithm

Key Generation : Generating randomly the key $z \in R_q$ by some given distribution χ.

Encryption : For any message m in the message space R_t, choosing a uniformly in R_q and choosing e in R_q randomly according to distribution χ. Then compute the ciphertext:

$$c = (a, c') = (a, az + te + m)$$

where z is the key and t is an integer.

Decryption : Using the key z, we can get the planetext:

$$m = (c' - az) \bmod t.$$

3.2 Matrix encryption algorithm

Let $\mathbf{M} = (m_{ij}) \in (R_t)^{h\times h}$ be a planetext matrix.

Key Generation : For any $j \in [h]$, choosing $z_j \in R_q$ randomly according to the distribution χ. Then the vector $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$ is the key vector.

Encryption : For any $i \in [h]$, choosing $a_i \in R_q$ randomly according to the distribution χ and we obtain the vector $\mathbf{a} = (a_1, a_2, \cdots, a_h)^T$. Construct the error matrix $\mathbf{E} = (e_{ij}) \in (R_q)^{h\times h}$, where each e_{ij} is chosen randomly in R_q. Then compute $c'_{ij} = a_i z_j + te_{ij} + m_{ij}$, where $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$ is the key vector. and construct the matrix $\mathbf{C}' = (c'_{ij})_{1\le i\le h, 1\le j\le h}$. Output the matrix $(\mathbf{a}, \mathbf{C}')$ as the ciphertext matrix, that is,

$$E_{\mathbf{z}}(\mathbf{M}; \mathbf{a}, \mathbf{E}) = \mathbf{C} = (\mathbf{a}, \mathbf{C}') = (\mathbf{a}, \mathbf{az}^T + t\mathbf{E} + \mathbf{M}).$$

Decryption : Using the key vector $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$, we can decrypt every element of the part $\mathbf{C}'$ of the ciphertext matrix $\mathbf{C}$, that is, $(c'_{ij} - a_i z_j) \bmod t = m_{ij}$. Then we can get the planetext matrix $\mathbf{M}$.

3.3 Lossy trapdoor functions

Injective functions generation Suppose that $\mathbf{I}$ is an $h \times h$ identity matrix, $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$ is the key vector, and the ciphertext matrix of the identity matrix $\mathbf{I}$ is $\mathbf{C} = (\mathbf{a}, \mathbf{C}')_{h\times(1+h)}$. Then $S_{inj}(\cdot)$ outputs $(\mathbf{C}, \mathbf{z})$, where $\mathbf{C}$ is the function index and $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$ is the key.

Lossy functions generation Suppose that $\mathbf{0}$ is an $h \times h$ zero matrix, $\mathbf{z} = (z_1, z_2, \cdots, z_h)^T$ is the key vector, and the ciphertext matrix of the zero matrix $\mathbf{0}$ is $\mathbf{C} = (\mathbf{a}, \mathbf{C}')_{h\times(1+h)}$. Then $S_{loss}(\cdot)$ outputs $(\mathbf{C}, \perp)$, where $\mathbf{C}$ is the function index.

Estimation algorithm $F_{ltdf}(\mathbf{C}, \mathbf{x})$, where $\mathbf{C}$ is the function index and $\mathbf{x} \in (R_t)^h$ is the input vector. Then the output is $\mathbf{y}^T = \mathbf{x}^T \cdot \mathbf{C}$.

Inversion algorithm For the injective function, we will compute $F_{ltdf}^{-1}(\mathbf{z}, \mathbf{y})$, where $\mathbf{z}$ is the key. Compute:

1. Denote $\mathbf{y}$ as $1 + h$ dimensional vector $(y_1, y_2, \cdots, y_h, y_{1+h})^T$;
2. Compute $x_i = (y_{i+1} - y_1 z_i) \bmod t$, where x_i is the i-th coordinate of $\mathbf{x}$ $(1 \le i \le h)$.

Then output $\mathbf{x} = (x_1, x_2, \cdots, x_h)$ as the preimage of $\mathbf{y}$.

Theorem 3.1 *Under the polynomial LWE assumption-Hermite Normal Form, that is, $PLWE_{f,q,\chi}$, the functions in above scheme are lossy trapdoor functions, whose leakage is $r = (n+h)\log_2 q - h\log_2 t$.*

Proof Firstly, the inversion algorithm $F^{-1}_{ltdf}(\mathbf{z}, \mathbf{y})$ is correct. For the function $\mathbf{y}^T = F_{ltdf}(\mathbf{C}, \mathbf{x}) = \mathbf{x}^T \cdot \mathbf{C}$, because $\mathbf{C} = (\mathbf{a}, \mathbf{C}') = (\mathbf{a}, \mathbf{a}\mathbf{z}^T + t\mathbf{E} + \mathbf{I})$, we have

$$y_{i+1} - y_1 z_i = x_i + t\sum_{l=1}^{h} x_l e_{li}.$$

Then $x_i = (y_{i+1} - y_1 z_i) \bmod t$ and $\mathbf{x} = (x_1, x_2, \cdots, x_h)^T$ is the preimage of $\mathbf{y}$.

Lossy functions and injective functions are indistinguishable, which is ensured by the security of the matrix encryption algorithm. For any two planetext matrices $\mathbf{M} = (m_{ij}) \in (R_t)^{h\times h}$ and $\mathbf{M}' = (m'_{ij}) \in (R_t)^{h\times h}$, the ciphertext matrices obtained by the matrix encryption algorithm are indistinguishable under the $PLWE_{f,q\chi}$ assumption. We will prove that for any planetext matrix $\mathbf{M} = (m_{ij}) \in (R_t)^{h\times h}$, the proper encryption $E_{\mathbf{z}}(\mathbf{M}; \mathbf{a}, \mathbf{E})$ and the uniform encryption $E_{\mathbf{z}}(\mathbf{M}; \mathbf{a}, \mathbf{R})$ of $\mathbf{M}$ are indistinguishable where the matrix $\mathbf{R} \leftarrow (R_q)^{h\times h}$ is chosen uniformly.

Using a standard hybrid argument, firstly define the hybrid experiments $H_0, H_1, \cdots, H_h$. In the experiment H_k, the output is the encrypted matrix $E_{\mathbf{z}}(\mathbf{M}; \mathbf{a}, \mathbf{E})$, where the first k-columns of "$\mathbf{E}$" is chosen according to the distribution χ and the other columns of "$\mathbf{E}$" is chosen uniformly. Thus, H_0 generates the uniformly encryption of the planetext matrix and H_h generates the proper encryption of the planetext matrix. We only need to prove that the experiments H_{k-1} and H_k are indistinguishable.

For any $k \in [h]$, we will consider the simulation algorithm S^O, where O generates samples according to the distribution $A_{\mathbf{z},t\chi}(\mathbf{z} \leftarrow (R_q)^h)$, that is, $\{(a_i, a_i z_j + te_{ij})\}_{i\in[h]\ and\ j\in[h]}$ where the elements z_j are chosen uniformly in R_q and the elements e_{ij} obey χ distribution, or the distribution $\{(a_i, u_i)\}_{i\in[h]}$ where the elements u_i are uniformly random on R_q. For $j \neq k$, S generates the key $z_j \leftarrow R_q$ independently. For $j = k$ and each $i \in [h]$, S queries the oracle O and has the samples (a_i, u_i). Let $\mathbf{a} = (a_1, a_2, \cdots, a_h)^T$ and $c'_{ik} = u_i + m_{ik}$. For the columns satisfying $j < k$ and all rows $i \in [h]$, S chooses the error $e_{ij} \leftarrow \chi$ independently; for all columns satisfying $j > k$ and all rows $i \in [h]$, S chooses the error $e_{ij} \leftarrow R_q$ uniformly. For all columns j $(j \neq k)$ and all rows $i \in [h]$, let $c'_{ij} = a_i z_j + te_{ij} + m_{ij}$. Then S outputs $(\mathbf{a}, \mathbf{C}')$.

If the sample (a_i, u_i) is uniform, the output of S is subjected to H_{k-1} because te_{ik} is chosen uniformly. If the sample (a_i, u_i) obeys $A_{\mathbf{z},t\chi}$, the output of S is subjected to H_k. Because $PLWE^{(l)}_{f,q,\chi}$ is hard and the Proposition 2.7, the distribution $A_{\mathbf{z},t\chi}$ and the uniform distribution are computely indistinguishable. Then the experiments H_{k-1} and H_k are computely indistinguishable.

Now we consider the leakage. The cardinality of the range is

$$|R_q| \times \left(\frac{q}{t}\right)^h = q^n \frac{q^h}{t^h} = \frac{q^{n+h}}{t^h},$$

and the cardinality of the domain is t^{nh}, so the residue leakage of the lossy trapdoor functions is

$$r = (n+h)\log_2 q - h\log_2 t.$$

Choosing n, h, t, q properly, the leakage phenomenon will be generated. □

3.4 Special cases

The conclusion we obtained in the above subsection is the general case. The above scheme has the following two special cases:

1. Let $t = 2$;
2. Let the input function be $\mathbf{x} \in (Z_t)^h$.

Based on the above lossy trapdoor functions, we can construct new all-but-one trapdoor functions.

4 New all-but-one trapdoor functions

Let $t = t(\lambda)$ be a large enough prime and R_t be a finite field. Let $\mathcal{B} = R_t$ be a collection of sets whose elements represent the branches. For any branch $b \in \mathcal{B}$, $\bar{\mathbf{b}}$ denotes the diagonal matrix corresponding to the branch b, that is,

$$\bar{\mathbf{b}} = \begin{pmatrix} b & 0 & \cdots & 0 \\ 0 & b & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b \end{pmatrix}.$$

The concrete construction of all-but-one trapdoor functions $\{S_{abo}, G_{abo}, G^{-1}_{abo}\}$ as follows:

– S_{abo} generates all-but-one trapdoor functions:
  Choose $b^* \in \mathcal{B}$ randomly as the lossy branch and call the matrix encryption algorithm to encrypt the matrix $\mathbf{M}^* = \overline{-\mathbf{b}^*}$, we can obtain the matrix $\mathbf{C}^* = E_{\mathbf{z}}(\mathbf{M}^*; \mathbf{a}, \mathbf{E})$, and then output the matrix index $\mathbf{C}^*$ and the trapdoor information $\mathbf{z}$;

– Estimation algorithm G_{abo}:
Input the branch $b \in \mathcal{B}$ and the function index $\mathbf{C}^*$, $G_{abo}(\mathbf{C}^*; b, \cdot)$ outputs a function $g_{(\mathbf{C}^*,b)}(\cdot) : R_t^h \to R_q^{h+1}$, where $\mathbf{x} \in (R_t)^h$ is the input vector and

$$\mathbf{y}^T = g_{(\mathbf{C}^*,b)}(\mathbf{x}) = \mathbf{x}^T \cdot (\mathbf{C}^* + (\mathbf{0}\ \mathbf{b}))$$

is the output vector, where $\mathbf{0}$ denotes h-dimensional zero vector;

– Inversive algorithm G_{abo}^{-1}:
Input the branch $b \in \mathcal{B}$ and the trapdoor information $(\mathbf{z}, b^*)$. G_{abo}^{-1} outputs the function $g^{-1}_{(\mathbf{C}^*,b)}(\cdot) : R_q^{h+1} \to R_t^h$. For $\mathbf{y} \in R_q^{h+1}$, decrypt $\mathbf{y}$ by decryption algorithm with the key $\mathbf{z}$ and then $\mathbf{y}'$ can be obtained. That is, $\mathbf{y}$ is denoted as $h+1$-dimensional vector $(y_1, y_2, \cdots, y_h, y_{h+1})^T$ and we compute $y_i' = (y_{i+1} - y_1 z_i) \bmod t (1 \leq i \leq h)$. So we can get the vector $\mathbf{y}' = (y_1', y_2', \cdots, y_h')^T$. We can compute $x_i = y_i' \cdot (b - b^*)^{-1} \bmod t$, and then $\mathbf{x} = (x_1, x_2, \cdots, x_h)^T$ is obtained. Thus

$$g^{-1}_{(\mathbf{C}^*,b)}(\mathbf{y}) = \mathbf{x}, \mathbf{y} \in R_q^{h+1}.$$

Theorem 4.1 *Under the polynomial LWE assumption-Hermite Normal Form, the functions in above scheme is (n, k)-all-but-one trapdoor functions, whose branch set is R_t and the leakage is $r = (n + h)\log_2 q - h \log_2 t$.*

The proof of this theorem is similar to that of the theorem 3.1.

5 Conclusions

In this paper,we proposed a new method of constructing lossy trapdoor functions and all-but-one trapdoor functions based on the polynomial learning with errors assumption. And our schemes have the following advantages compared with the previous lattice-based constructions of lossy trapdoor functions:

Our schemes have small function index parameters as shown in Table 1. In the scheme of [13], if the input needs k bits, the function index will need $2k^2 \log q_1$ bits and the key needs $k \log q_1$ bits. While in our scheme, if the input needs $nh \log t$ bits, the function index will need $h(h+1)n \log q$ bits and the key needs $nh \log q$ bits. Easy to find that choosing h and t properly, then the needed bits of the function index and key corresponding to average single input bit is smaller than that of the scheme in [13].

Our schemes have high speed of estimation and inversion by choosing the parameters properly as shown in Table 2. And they use the ring operation, thus they are realized with high speed. Moreover, our schemes have smaller complexity and the higher efficiency. The rounding complexity of the scheme in [13] is $O(n^2\omega(1))$. While our scheme did not use counting technology, the rounding complexity of our scheme is zero.

Our functions have large throughput as shown in Table 3. The throughput of the base encryption algorithm in [13] is $\log p$ bits and the trapdoor function input has n bits, while those in our schemes are $n \log t$ bits and $nh \log t$ bits respectively, where R_t is the plaintext space.

Table 1 The comparison on function index parameters

(Bits)	Function index	Key
[PW11] (input k)	$2k^2 \log q_1$	$k \log q_1$
Ours(input $nh \log t$)	$h(h+1)n \log q$	$nh \log q$

Table 2 The comparison on rounding complexity

	Runding complexity
[PW11]	$O(n^2\omega(1))$
Our scheme	0

Table 3 The comparison on throughput

	Throughput	TDs input
[PW11]	$\log p$ bits	n bits
Our scheme	$n \log t$ bits	$nh \log t$ bits

Acknowledgements This work is supported by the National Science Foundation of China under Grant 61373171 and the 111 Project under Grant B08038.

References

1. Lehman, R.S.: Factoring large integers. Math. Comput. **28**(126), 637–646 (1974)
2. Wagstaff, S.S., Smith, J.W.: Methods of factoring large integers. Lect. Notes Math. **1240**, 261–303 (1987)
3. Silverman, R.D.: Massively distributed computing and factoring large integers. Commun. ACM. **34**(11), 95–103 (1991)
4. Shamir, A.: Factoring large numbers with the TWINKLE device. In: Koc, C.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems, pp. 727–727. Springer, Berlin (1999)
5. Boneh, D.: Twenty years of attacks on the RSA cryptosystem. Not. AMS. **46**(2), 203–213 (1999)
6. Miller, V.S.: Use of elliptic curves in cryptography. In: Conference on the Theory and Application of Cryptographic Techniques, pp. 417–426. Springer, Berlin (1985)
7. Maurer, U.: Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In: Desmedt, Y.G. (ed.) Advances in cryptology—CRYPTO94, pp. 271–281. Springer, Berlin (1994)
8. Shoup, V.: Lower bounds for discrete logarithms and related problems. Eurocrypt **97**, 256–266 (1997)

9. Ajtai, M.: Generating hard instances of lattice problems. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing. ACM, pp. 99–108 (1996)
10. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. ACM, pp. 601–610 (2001)
11. Kuznetsov, S.O.: On computing the size of a lattice and related decision problems. Order **18**(4), 313–321 (2001)
12. Pujol, X., Stehl, D.: Solving the shortest lattice vector problem in time 22.465 n. IACR Cryptol. ePrint. Arch. **2005**, 605 (2009)
13. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)
14. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011)
15. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology—EUROCRYPT 2012. Springer, Berlin, pp. 700–718 (2012)
16. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) Automata, Languages and Programming. Springer, Berlin, pp. 1–9 (1999)
17. Cheng, S., Nguyen, K., Wang, H.: Policy-based signature scheme from lattices. Des. Codes Cryptogr. **81**(1), 1–32 (2015)
18. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing. ACM **2008**, 197–206 (2008)
19. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. J. ACM. **60**(6), 43 (2013)
20. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology—EUROCRYPT 2013. Springer, Berlin, pp. 35–54 (2013)
21. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. SIAM J. Comput. **39**(7), 3058–3088 (2010)
22. Alwen, J., Krenn, S., Pietrzak, K., et al.: Learning with rounding, revisited. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology—CRYPTO 2013. Springer, Berlin, pp. 57–74 (2013)
23. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent message. In: Rogaway, P. (ed.) Advances in Cryptology—CRYPTO 2011. Springer, Berlin, pp. 505–524 (2011)

Chengli Zhang received the B.S. degree in Mathematics and Applied Mathematics from Harbin Normal University, China in 2007, and M.S. degree in Applied Mathematics from Xi'an University of Architecture and Technology, China in 2011. She is currently pursuing toward the Ph.D. degree in Telecommunications Engineering with the State Key Laboratory of Integrated Service Networks, Xidian University, China. Her research interests focus on the areas of information security, public key cryptography, and semigroup theory.

Wenping Ma received the B.S. and the M.S. degrees from Shaanxi Normal University in 1987 and 1990, respectively, and the Ph.D. degrees from Xidian University in 1999. Since 2000, he has been the professor in the school of Telecommunications Engineering, Xidian University, Xi'an Shaanxi, China. His research interests include information theory, communication theory, error-correcting codes and information security. He is a senior member of Chinese Institute of Electronics, a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and Chinese Communication Society.

Hefeng Chen received the B.S. degree and M.S. degree in mathematics and applied mathematics from School of Mathematical Sciences, Xiamen University in 2005 and 2008, respectively, and the Ph.D. degrees in cryptography, from School of Telecommunications Engineering from Xidian University in 2016. Since 2016, she has been the lecturer in Computer Engineering College, Jimei University, Xiamen Fujian, China. Her research interests focus on the areas of information security, public key cryptography.

Feifei Zhao is currently working toward the Ph.D. degree in military communication with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China. Her main research interests include the application of optimization to resource allocation and interference management problems in wireless networks. Now she is working on the topic of resource management in heterogeneous networks and graph theory for wireless networks.