information

Article

Continuous Leakage Resilient Lossy Trapdoor Functions

Sujuan Li [1,*], Yi Mu [2], Mingwu Zhang [3] and Futai Zhang [4]

[1] School of Mathematical and Physical Sciences, Nanjing Tech University, Nanjing, 211800, China
[2] School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia; ymu@uow.edu.au
[3] School of Computer Science, Hubei University of Technology, Wuhan, 430068, China; csmwzhang@gmail.com
[4] School of Computer Science and Technology, Nanjing Normal University, Nanjing, 210023, China; zhangfutai@njnu.edu.cn
[*] Correspondence: lisujuan1978@126.com; Tel.: +86-25-58139527

Academic Editors: Man Ho Au, Jinguang Han, Joseph K. Liu and Ali El Kaafarani
Received: 5 February 2017; Accepted: 17 March 2017; Published: 23 March 2017

Abstract: Lossy trapdoor functions (LTFs) were first introduced by Peikert and Waters (STOC'08). Since their introduction, lossy trapdoor functions have found numerous applications. They can be used as tools to construct important cryptographic primitives such as injective one-way trapdoor functions, chosen-ciphertext-secure public key encryptions, deterministic encryptions, et al. In this paper, we focus on the lossy trapdoor functions in the presence of continuous leakage. We introduce the new notion of updatable lossy trapdoor functions (ULTFs) and give their formal definition and security properties. Based on these, we extend the security model to the LTFs against continuous leakage when the evaluation algorithm is leakage resilient. Under the standard DDH assumption and DCR assumption, respectively, we show two explicit lossy trapdoor functions against continuous leakage in the standard model. In these schemes, using the technology of matrix kernel, the trapdoor can be refreshed at regular intervals and the adversaries can learn unbounded leakage information on the trapdoor along the whole system life. At the same time, we also show the performance of the proposed schemes compared with the known existing continuous leakage resilient lossy trapdoor functions.

Keywords: lossy trapdoor functions; continuous leakage; ULTFs; DDH; DCR

1. Introduction

Lossy trapdoor functions (LTFs) were firstly introduced by Peikert and Waters (STOC 2008) [1]. A collection of lossy trapdoor functions can be divided into two computationally indistinguishable families. The first family is the injective functions which can be efficiently inverted using a trapdoor. The other family is the lossy functions under which the image size of these functions is significantly smaller than the size of their domain. Hence, the lossy functions loose a lot of information about their input. Additionally, injective and lossy functions are efficiently samplable.

Since their introduction, lossy trapdoor functions have found numerous applications. It can be used as a tool to construct important cryptographic primitives such as injective one-way trapdoor functions, chosen plaintext secure (CPA) and chosen ciphertext secure (CCA) public key encryptions (PKE) in the standard model and oblivious transfer (OT). In addition, LTFs have already found various other applications, including deterministic PKE schemes [2,3], OAEP-based PKE schemes, "hedged" PKE schemes for protecting against bad randomness [4], selective opening attack (SOA) secure PKE scheme [5] and efficient non-interactive string commitments [6].

Leakage-resilient cryptographic systems have received a lot of attention in recent years. The feature of a leakage resilient cryptosystem is that it remains secure even when some secret internal information, including the secret key, is leaked to the adversary. In the traditional security analysis, security models treat such internal information as perfectly hidden from the adversary. With the development of various side-channel attacks, it is clear that the traditional view is inconsistent with some physical realities [7]. To stand against such attacks, cryptographic researchers have paid much to the design of leakage-resilient cryptosystems [8–14].

The continuous leakage resilient (CLR) model was introduced by Dodis et al. [15] and Brakerski et al. [16]. It is a more powerful security model since it allows the adversary to learn unbounded leakage on the system's secret memory during the main operation of the system. There are a variety of CLR schemes, including CLR one-way relations [15,17], CLR probability PKE [16–18], CLR Identity-based encryption (IBE) [16,19], CLR secure multiparty computation [20], CLR interactive proofs [21], CLR signatures, CLR identification schemes and CLR authenticated key agreement protocols [15].

To withstand continuous leakage, the secret key must be continuously refreshed requiring that: (1) the functionality of the cryptosystem is preserved even after updating the keys an arbitrary number of times; (2) one can not combine the leaked values from different versions of the secret key to break the system. Such a model of invisible key updates was formalized by Alwen et al. [22], where one assumes that there exists a trusted and leak-free device who uses some updatable key uk to continuously refresh the secret key in a way that still satisfies the above two requirements. The leak-free device is only present during the key updates, but not during the normal operations just like decryption when the leakage actually happens. In [17], they informally refer to this CLR model of invisible key updates as the floppy model where there is assumed an external leak-free storage that is only present for refreshing operations.

1.1. Our Motivation

Lossy trapdoor functions play an important role in public key cryptosystems. Its special construction and properties decide that it is the building block of the cryptosystems. As we all know, the CLR model is the most demanding security model in the cryptosystem. Therefore, designing lossy trapdoor functions against continuous leakage is an interesting and practical topic.

Based on the work of Brakerski et al. [16], Koppula et al. [23] firstly gave the security model of lossy trapdoor functions under continuous leakage and presented the lossy trapdoor functions against continuous leakage, which is a base of the deterministic public key encryption against continuous leakage. Their security model is mainly based on the all-but-one (ABO) LTFs of Peikert and Waters in [1]. Under this model, their proposal is not concise and efficient in which they utilized many bi-linear parings to encrypt only one bit. Hence, their LTFs against continuous leakage is so complicated that it can not be used in practice efficiently. Qin and Liu et al. first introduced the leakage resilient lossy trapdoor functions [24]. In their work, the structure of LTFs is slightly different from the one introduced by Peikert and Waters in [1]. In [1], the evaluation key of a LTF includes the public parameters. However, in [24], they distinguish between the public parameters and the evaluation key with two independent algorithms. However, the slight change on the constructure did not have an influence on their scheme to satisfy the security properties of LTFs.

Motivated by the work of Qin and Liu et al. [24], we focus on how to construct efficient and practical LTFs against continuous leakage in the floppy model.

1.2. Our Contribution

In this work, our contribution is described as follows:

1. We introduce the new notion of updatable lossy trapdoor functions (ULTFs) based on the LTF structure of [24], where the key sample algorithm is divided into two independent steps. In the first algorithm, it takes in the security parameter 1^κ and outputs a public parameter pp and the

trapdoor td; in the second algorithm, it takes in pp and injective/lossy parameter $b \in \{0, 1\}$ and outputs the injective/lossy evaluation key ek, which is related to b. At the same time, we also give the security requirements such as the indistinguishability of injective/lossy evaluation key, etc. When the evaluation algorithm F is leakage resilient, we can achieve the LTFs against continuous leakage, which we denote as CLR-LTFs for short. With the help of the new notion of ULTFs, we achieve the security model of CLR-LTFs in the floppy model. When the adversary is equipped with the public parameter and additional information from the leakage oracle during each time period, it still is not able to distinguish the injective and lossy evaluation keys.

2. Based on the ElGamal-like PKE scheme in vector form [17,25,26], which is additively homomorphic and CPA-secure against continuous leakage, we achieve two proposals of CLR-LTFs under the standard Decisional Diffie–Hellman (DDH) and Decisional Composite Residuosity (DCR) assumptions, respectively. In the two CLR-LTF schemes, with the public parameters and the evaluation key fixed, we utilize the technology of the matrix kernel to complete the refreshment of the trapdoor. Our first proposal is obtained by embedding the CLR ElGamal-like PKE scheme into the matrix-based LTFs of [1] n times, where the ciphertexts constitute the rows of the matrix R and the columns of the matrix Q, respectively. Through the n-time expansion of the secret key of a single ElGamal-like PKE scheme, the leakage rate of the achieved CLR-LTF is decreased from $1 - o(1)$ into $\frac{1}{n}$ for maintaining the indistinguishability of the injective or lossy evaluation keys. In order to improve the leakage rate in each time period, we extend the group from a prime order group to a composite order group and get the second CLR-LTFs based on the DCR assumption, where the leakage rate can arrive at 1.
3. Compared with the other known CLR-LTFs constructions introduced by Koppula et al. [23], we give an efficiency comparison as below (Table 1).

Table 1. Efficiency comparison.

Scheme	Hardness Assumption	Leakage Rate	$\lvert m \rvert$	Pairing	Group
[23]	DDH	$1/2$	1-bit	Yes	Prime order
[23]	SXDH	$1 - o(1)$	1-bit	Yes	Prime order
Ours	DDH	$1/n$	n-bit	No	Prime order
Ours	DCR	$1 - o(1)$	$\alpha \log N$-bit	No	Composite order

$|m|$ denotes the length of the encrypted massage; $n \approx \Theta(\kappa)$ where κ is the security parameter; N is an RSA modulus which will be explained by detail in Section 6; $\alpha \geq 1$ is a nature number; DDH means Decisional Diffie-Hellman assumption; SXDH means Symmetric External Diffie-Hellman assumption; DCR means Decisional Composite Residuosity assumption.

1.3. Organization

The rest of the article is organized as follows. In Section 2, we review some preliminaries which would be used in this paper. In Section 3, we introduce the new notion of updatable lossy trapdoor functions and present the formal definition and security properties. Meanwhile, we extend the security model to continuous leakage. Next, we introduce the CLR ElGamal-like PKE scheme with some important security properties which will be borrowed for the following concrete CLR-LTFs in Section 4. Then, we present two explicit CLR-LTFs. The first CLR-LTF under the DDH assumption in the prime order group is shown in Section 5. The second CLR-LTF under the DCR assumption in the composite order group is presented in Section 6, respectively. We also prove that these schemes are satisfying the security properties that have been given in Section 4. Lastly, we get a conclusion and direct the future work in Section 7.

Notion:
$\mathrm{negl}(\kappa)$ is negligible function with security parameter κ;
$[t]$ denotes the set $\{1, 2, \cdots, t\}$, where t is a natural number;
$\log x$ denotes the discrete logarithm of x in the base 2;
$\mathsf{Rk}_i(\mathbb{Z}_p^{n\times m})$ denotes the uniform distribution on any n-by-m matrices over $\mathbb{Z}_p$ of rank i.

2. Preliminaries

In this section, we present some basic tools that will be used in our constructions and security proofs. We formally state some decisional assumptions and present some results about the leftover hash lemma.

2.1. Decisional Assumptions

2.1.1. Decisional Diffie–Hellman (DDH) Assumption

We assume a probability polynomial time (PPT) algorithm $\mathcal{G}(1^\kappa)$ which takes as input 1^κ and outputs a tuple of $\mathbb{G} = (G, p, g)$, where G is a cyclic group of prime order p and g is a generation of G. The Decisional Diffie–Hellman (DDH) assumption holds iff

$$\mathrm{Adv}_{G,\mathcal{A}}^{\mathrm{DDH}} := |\Pr[\mathcal{A}(g_1, g_2, g_1^r, g_2^r) = 1] - \Pr[\mathcal{A}(g_1, g_2, g_1^r, g_2^{r'}) = 1]| \leq \mathrm{negl}(\kappa)$$

for any PPT adversary $\mathcal{A}$, where $g_1, g_2 \in G$ and $r \in \mathbb{Z}_q$, $r' \in \mathbb{Z}_q \setminus \{r\}$.

We can extend the standard DDH assumption to the following form. For a group (G, p, g) and random elements $g_1, g_2, \cdots, g_l \in G$, we define the two sets:

$$L := \{(g_1^r, g_2^r, \cdots, g_l^r) : r \in \mathbb{Z}_p\};$$
$$X := \{(g_1^{r_1}, g_2^{r_2}, \cdots, g_l^{r_l}) : r_1, r_2, \cdots, r_l \in \mathbb{Z}_p\}.$$

If $x \in L$, the corresponding r is called a witness for x. At the same time, (X, L) forms a subset membership problem [26] whose hardness is subject to the DDH assumption [25].

On the other hand, Ref. [26] showed that the DDH assumption is equivalent to the assumption that it is hard to distinguish between an n-by-m matrix X with rank $i \geq 1$ and one with rank $j > i$ in the exponent of a generator g of a prime order group G.

2.1.2. Rank Hiding Assumption

Following the parameters of the DDH assumption, let $\mathsf{Rk}_i(\mathbb{Z}_p^{n\times m})$ denote the uniform distribution on all n-by-m matrices over $\mathbb{Z}_p$ of rank i. The rank hiding assumption [17] holds iff

$$\begin{aligned}\mathrm{Adv}_{G,\mathcal{A}}^{\mathrm{rh}} := &|\Pr[\mathcal{A}((G, p, g, g^X) : X \leftarrow \mathsf{Rk}_i(\mathbb{Z}_p^{n\times m})) = 1] \\ &-\Pr[\mathcal{A}((G, p, g, g^X) : X \leftarrow \mathsf{Rk}_j(\mathbb{Z}_p^{n\times m})) = 1]| \leq \mathrm{negl}(\kappa)\end{aligned}$$

for any PPT adversary $\mathcal{A}$.

2.1.3. Extended Rank Hiding Assumption

Based on the rank hiding assumption, the extended rank hiding assumption [17] states that, for any PPT adversary $\mathcal{A}$, the advantage

$$\begin{aligned}\mathrm{Adv}_{G,\mathcal{A}}^{\mathrm{erh}} := &|\Pr[\mathcal{A}((G, p, g, g^X, v_1, \cdots, v_t) : X \leftarrow \mathsf{Rk}_i(\mathbb{Z}_p^{n\times m}); \{v_l\}_{l=1}^t \in \mathsf{kernel}(X)) = 1] \\ &- \Pr[\mathcal{A}((G, p, g, g^X, v_1, \cdots, v_t) : X \leftarrow \mathsf{Rk}_j(\mathbb{Z}_p^{n\times m}); \{v_l\}_{l=1}^t \in \mathsf{kernel}(X)) = 1]| \leq \mathrm{negl}(\kappa),\end{aligned}$$

where $m, n \in \mathbb{N}, j > i \in \mathbb{N}$ and $t \leq \min\{n, m\} - \max\{i, j\}$.

2.1.4. Decisional Composite Residuosity (DCR) Assumption

We assume a group $Z^*_{N^{\alpha+1}}$ is a multiplicative group where $s \geq 1$ is an integer. In addition, the integer $N = PQ$ is an RSA modulus, which means that P and Q are odd primes of equivalent bit length. The decisional composite residuosity (DCR) assumption holds on the group $Z^*_{N^{\alpha+1}}$ iff

$$\mathrm{Adv}^{DCR}_{N,\mathcal{A}} := |\Pr[\mathcal{A}(N,g) = 1] - \Pr[\mathcal{A}(N, g \cdot T) = 1]| \leq \mathrm{negl}(\kappa)$$

for any PPT adversary $\mathcal{A}$, where $g \in \mathsf{G}$ is chosen at random (where G is a cyclic group of order N^{α}) and $T := 1 + N(\mathrm{mod} N^{\alpha+1})$.

2.2. Generalized Leftover Hash Lemma

The statistical distance between two random variables X and Y over a finite domain Ω is $\mathrm{SD}(X,Y)$=$\frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. We write $X \approx_{\epsilon} Y$ to denote $\mathrm{SD}(X,Y)$$\leq \epsilon$, and $X \approx Y$ to denote that the statistical distance is negligible. The min-entropy of a random variable X is $H_{\infty}(X) = -\log(max_x \Pr[X = x])$.

We use the notion of average min-entropy, which captures the remaining unpredictability of a random variable X conditioned on another random variable Y, formally defined as:

$$\widetilde{H}_{\infty}(X|Y) = -log(E_{y\in Y}[2^{-H_{\infty}(X|Y=y)}]),$$

where $E_{y\in Y}$ denotes the expected value over all values of Y.

Lemma 1 [27]**.** *For any random variables X, Y, Z, if Y has 2^r possible values, then*

$$\widetilde{H}_{\infty}(X|(Y,Z)) \geq \widetilde{H}_{\infty}(X|Z) - r.$$

In particular,

$$\widetilde{H}_{\infty}(X|Y) \geq H_{\infty}(X) - r.$$

Definition 1 [27]**.** *A function $\mathsf{Ext} : \mathcal{X} \times \{0,1\}^t \to \mathcal{Y}$ is an average-case (m,ϵ)-strong extractor if, for all pairs of random variables (X,Z) such that $X \in \mathcal{X}$ and $\widetilde{H}_{\infty}(X|Z) \geq m$, it holds that*

$$\mathrm{SD}((\mathsf{Ext}(X,S), S, Z), (U_{\mathcal{Y}}, S, Z)) \leq \epsilon,$$

where S is uniform in $\{0,1\}^t$ and $U_{\mathcal{Y}}$ is uniform over $\mathcal{Y}$.

Definition 2 (Universal Hashing). *A family $\mathcal{H}$, consisting of deterministic functions $h : \mathcal{X} \to \mathcal{Y}$, is a universal hash family if, for any $x_1 \neq x_2 \in \mathcal{X}$, we have $Pr_{h\leftarrow\mathcal{H}}[h(x_1) = h(x_2)] \leq 1/|\mathcal{Y}|$.*

Lemma 2 (Generalized Leftover Hash Lemma) [27]**.** *Assume that the family $\mathcal{H} = \{H_k : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is a universal hash family. Then, for any two random variables X, Z and $k \in \mathcal{K}$, it holds that*

$$\mathrm{SD}((H_k(X), k, Z), (U_{\mathcal{Y}}, k, Z)) \leq \frac{1}{2}\sqrt{2^{-\tilde{H}_{\infty}(X|Z)}|\mathcal{Y}|}.$$

This lemma implies that any universal hash functions are good extractors. For two random variables X and Y, a family of universal hash functions $\{H_k : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is an average-case (m,ϵ)-strong extractor $\mathrm{Ext} : \mathcal{X} \times \mathcal{K} \to \mathcal{Y}$ as long as $\tilde{H}_{\infty}(X|Z) \geq m$ and $\log|\mathcal{Y}| \leq m - 2\log(1/\epsilon) + 2$.

3. Updatable Lossy Trapdoor Function

In this section, we will introduce the new notion of updatable lossy trapdoor functions (ULTFs). Though Koppula et al. [23] has introduced a notion of LTFs resilient to continual memory leakage, their notion was mainly based on the all-but-one (ABO) LTFs of Peikert and Waters in [1]. The new notion, which will be presented as follows, is mainly based on the LTFs structure of Qin and Liu et al. [24], which is slightly different from the one introduced by Peikert and Waters in [1]. In [1], the evaluation key of a LTF includes the public parameters. However, in [24], they distinguish between the public parameters and the evaluation key with two independent algorithms. As a result, the change in the structure does not have any influence on the security. Based on the new notion, we can extend the ULTFs to CLR-LTFs naturally when the evaluation algorithm is leakage resilient.

3.1. Definition of Updatable Lossy Trapdoor Functions

At first, we give some related functions about the security parameter κ:

$d(\kappa)$: the inpute lenghth of the polynomial about κ;
$k(\kappa)$: the lossiness $k(\kappa) \leq d(\kappa)$.

Now, we introduce the new notion of updatable lossy trapdoor functions.

Definition 3 (Updatable Lossy Trapdoor Functions). *A collection of updatable (d, k)-lossy trapdoor functions is a 5-tuple of (possible probabilistic) polynomial-time algorithms (PTAs) (G, S, F, F^{-1}, U) such that:*

1. ***Public Parameter.*** *$G(1^\kappa)$: It is a probabilistic PTA which takes in the security parameter 1^κ and outputs the public parameter and the trapdoor (pp, td).*
2. ***Public Parameter.*** *$S(\mathsf{pp}, b)$: It is a probabilistic PTA which takes in the public parameter pp and $b \in \{0, 1\}$ and samples an evaluation key ek which is also called the function index.*
3. ***Evaluation.*** *$F(ek, x)$: It is a deterministic PTA which takes in the evaluation key ek and $x \in \{0, 1\}^d$ and outputs the image y.*
4. ***Inversion.*** *$F^{-1}(td, y)$: It is a deterministic PTA which takes in the image y and the trapdoor td and outputs $x \in \{0, 1\}^d$ or $\perp$.*
5. ***Update.*** *$U(uk, td)$: It is a probabilistic PTA which takes in the updatable key uk and the original trapdoor td and outputs the updated trapdoor td' such that $|td| = |td'|$.*

3.2. Basic Properties

We require that the ULTF $(\mathsf{G}, \mathsf{S}, \mathsf{F}, \mathsf{F}^{-1}, \mathsf{U})$ has some basic properties, indicating its correctness an hardness requirements:

- Correctness. For all $(\mathsf{PP}, td) \leftarrow \mathsf{G}(1^\kappa)$, all $ek \leftarrow \mathsf{S}(\mathsf{pp}, 1)$ and all $x \in \{0, 1\}^d$, it holds that $\mathsf{F}^{-1}(td, \mathsf{F}(ek, x)) = x$, which is the preimage of y. On the other hand, it requires that, with the fixed public parameter pp and the evaluation key ek, the updated trapdoor td' can also recover the preimage x of y correctly in the injective mode, i.e., it holds that $\mathsf{F}^{-1}(td', \mathsf{F}(ek, x)) = x$.
- Injective/Lossy. For the third evaluation algorithm $\mathsf{F}(ek, \cdot)$, it requires that, for any $ek \leftarrow \mathsf{S}(\mathsf{pp}, 1)$, the function $\mathsf{F}(ek, \cdot)$ is in the injective mode; and for any $ek \leftarrow \mathsf{S}(\mathsf{pp}, 0)$ the function $\mathsf{F}(ek, \cdot)$ is in the lossy mode. The image size of the lossy function $\mathsf{F}(ek, x)$ is at most 2^{d-k}. Even when the evaluation $\mathsf{F}(ek, x)$ is in the injective mode, it requires that it can be inverted to the correct preimage using either the trapdoor td or any of its polynomial frequency updated trapdoor td'.
- Indistinguishability. For the second public parameter algorithm $\mathsf{S}(\mathsf{pp}, b)$, the two evaluation keys ek respectively produced by $\mathsf{S}(\mathsf{pp}, 1)$ and $\mathsf{S}(\mathsf{pp}, 0)$ are computationally indistinguishable even after the trapdoor updates.

3.3. Extension

For the particular structure, the ULTFs can be viewed as a special lossy trapdoor function which served as a fundamental tool in constructing cryptographic primitives in both leakage-free and leaky settings. Here, if we combine the ULTF with the leakage property efficiently, we can achieve the continuous leakage resilient (CLR) LTFs. Based on the new notion of ULTFs, we give the security model of the CLR-LTFs as follows.

We consider the security model in the floppy model [17]. This means that during the trapdoor update, there is leak-free device available and between two trapdoor updates there is bounded leakage about the trapdoor (see [17] for more details).

Definition 4 (Lossy Trapdoor Functions against Continuous Leakage). *We say that ULTFs* (G, S, F, F^{-1}, U) *is a collection of continuous* λ*-bit (weak) leakage resilient* (d, k)*-LTFs (denoted* λ*-CLR-LTFs) in the floppy model if the ULTFs satisfy the basic properties above, and, for any PPT* λ*-key leakage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the advantage*

$$\mathrm{Adv}^{\lambda-\mathrm{CLR}}_{\mathrm{ULTF},\mathcal{A}}(\kappa) := |\Pr[\mathrm{Exp}^{\lambda-\mathrm{CLR}}_{\mathrm{ULTF},\mathcal{A}}(\kappa, 0) = 1] - \mathrm{Exp}^{\lambda-\mathrm{CLR}}_{\mathrm{ULTF},\mathcal{A}}(\kappa, 1) = 1]| \le \mathrm{negl}(\kappa),$$

where the experiment $\mathrm{Exp}^{\lambda-\mathrm{CLR}}_{\mathrm{ULTF},\mathcal{A}}(\kappa, \gamma)$ $(\gamma \in \{0, 1\})$ is described as:

```
Experiment $\mathrm{Exp}^{\lambda-\mathrm{CLR}}_{\mathrm{ULTF},\mathcal{A}}(\kappa, \gamma)$:
    $(\mathsf{pp}, td_0) \leftarrow \mathsf{G}(1^\kappa)$
    For $i = 0, 1, 2, \cdots, t$, where $t$ is polynomial in the security parameter $\kappa$
    $\{State_i \leftarrow \mathcal{A}_1^{\mathrm{leakage}(td_i)}(\mathsf{pp})$, where $|\mathrm{leakage}(td_i)| \le \lambda$
    $td_{i+1} \leftarrow \mathsf{U}(uk, td_i)\}$, where $uk$ is the update key
    $ek \leftarrow \mathsf{S}(\mathsf{pp}, \gamma)$
    $\gamma' \leftarrow \mathcal{A}_2(State_{i \in [t]}, ek)$
    output $\gamma'$.
```

Remark 1. *In this security model, the adversary is only allowed to obtain leakage before it can see the evaluation key ek; therefore, the security of CLR-LTF in this paper is weak key leakage.*

4. ElGamal-Like Public Key Encryption Scheme

Briefly, we introduce the ElGamal-like Encryption scheme which will be elegantly embedded into the following continuous leakage resilient LTFs. In addition, we will utilize some good algebraic properties of this cryptographic structure in the following. For the security parameter, κ, $\mathbb{G} = (G, p, g) \leftarrow \mathcal{G}(1^\kappa)$. The scheme is run in group G with prime order p, for some negligible $\epsilon = \epsilon(\kappa)$ set $l = 2 + \frac{\lambda + 2\log(1/\epsilon) - 2}{\log p}$. The ElGamal-like PKE (KeyGen,Encrypt,Decrypt) is operated as follows:

1. $\mathsf{KeyGen}(1^\kappa)$: Run $\mathbb{G} = (G, p, g) \leftarrow \mathcal{G}(1^\kappa)$. Choose vector $\boldsymbol{w} \in \mathbb{Z}_p^l$ and $\mathbf{s} \in \mathbb{Z}_p^l$ and let $h = g^{\langle \boldsymbol{w}, \boldsymbol{s} \rangle} \in G$. The public key is $pk = (G, p, g, g^{\boldsymbol{w}}, h)$. The secret key is set to $sk = \boldsymbol{s}$.
2. $\mathsf{Encrypt}(pk, m)$: Given a public key $pk = (G, p, g, g^{\boldsymbol{w}}, h)$ along with a message $m \in G$, pick a random scalar $r \in \mathbb{Z}_q$ uniformly at random and output the ciphertext $c = (c_1, c_2) = (g^{r\boldsymbol{w}}, h^r \cdot m)$.
3. $\mathsf{Decrypt}(sk, c)$: Given a ciphertext $c = (c_1, c_2)$ along with a secret key $sk = \mathbf{s}$ output $m = c_2 \cdot c_1^{-\boldsymbol{s}}$.

The correctness holds directly with $h^r = g^{r\langle \boldsymbol{w}, \boldsymbol{s} \rangle} = g^{\langle r\boldsymbol{w}, \boldsymbol{s} \rangle}$. Evidently, the above scheme is a variant of the ElGamal public key encryption in vector form. On the other hand, it also can been seen as the BHHO (Boneh, Halevi, Hamburg, Ostrovsky) public key encryption [25] when $\boldsymbol{s} \in \{0, 1\}^n$. As we all know, this primitive has some good cryptographic properties. We will use these properties in our LTFs against continuous key leakage.

From the leakage resilient aspect, Ref. [25,26] showed that, given the public key and any λ bits of leakage, $\tilde{H}(sk|(pk, \lambda)) \ge \log p + 2\log(1/\epsilon) - 2$. The leftover hash lemma provides that, with overwhelming probability over the choice of $c_1 \in X \setminus L$, it holds that h^r is ϵ-close to the uniform distribution over G.

Lemma 3. *If the DDH assumption is hard in the p-prime order group G, then the above scheme is a* λ*-LR-CPA secure PKE scheme as long as the leakage parameter* $\lambda \le (l - 2)\log(p) - 2\log(1/\epsilon) + 2$*, where* $\epsilon = \epsilon(\kappa)$ *is some negligible function about the security parameter* κ*.*

From the continuous leakage resilient aspect, Ref. [17] showed that, with the updated key $\boldsymbol{w} \in \mathbb{Z}_p^l$, we can update the secret key with $sk' = sk + \boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathsf{kernal}(\boldsymbol{w})$. With the fixed public key, the updated key sk' can also decrypt the ciphertext correctly. Combined with the above lemma, with the help of the (extended) rank hiding assumption, the above scheme is a λ-CLR-CPA secure PKE scheme.

Lemma 4. *Under the extended rank hinging assumption and the DDH assumption for $\mathcal{G}$, then the above scheme is a λ-CLR-CPA secure PKE scheme in the floppy model as long as the leakage parameter $\lambda \leq (l-2)\log(p) - 2\log(1/\epsilon) + 2$, where $\epsilon = \epsilon(\kappa)$ is some negligible function about the security parameter κ.*

5. Continuous Leakage Resilient LTF from the DDH Assumption

In this section, based on the ElGamal-like Encryption scheme, we show a lossy trapdoor function against continuous trapdoor leakage.

5.1. The Scheme

In this section, we show how to construct continuous leakage resilient lossy trapdoor function (CLR-LTF) from the continuous leakage resilient CPA-secure ElGamal-like PKE.

For some negligible $\epsilon = \epsilon(\kappa)$ set, $l = 2 + \frac{\lambda + 2\log(1/\epsilon) - 2}{\log p}$. The construction CLR-TDF=(G, S, F, F^{-1}, U) is presented as follows:

1. $\mathsf{G}(1^\kappa)$: Run $\mathbb{G} = (G, p, g) \leftarrow \mathcal{G}(1^\kappa)$. Choose $g_1 = g^{w_1}, g_2 = g^{w_2}, \cdots, g_l = g^{w_l} \in G$ and let $\boldsymbol{w} = (w_1, w_2, \cdots, w_l) \in \mathbb{Z}_p^l$, then $g^{\boldsymbol{w}} = (g_1, g_2, \cdots, g_l)$. Choose n tuples of secret keys $\boldsymbol{s}_i = (s_{i1}, s_{i2}, \cdots s_{il}) \in \mathbb{Z}_p^l$ for $i \in [n]$. Let $h_i = \Pi_{j=1}^l g_j^{s_{ij}} = g^{\langle \boldsymbol{w}, \boldsymbol{s}_i \rangle}$. Output
$$\mathsf{pp} = (G, p, g, g^{\boldsymbol{w}}, h_1, h_2, \cdots, h_n), td = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n), uk = \boldsymbol{w}.$$
2. $\mathsf{S}(\mathsf{pp}, b)$: Given $b \in \{0, 1\}$. For $i \in [n]$, let $\boldsymbol{R}_i = (g_1^{r_i}, g_2^{r_i}, \cdots, g_l^{r_i}) \in L$ with a witness $r_i \in \mathbb{Z}_p$ independently at random.
Let $R = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{R}_2 \\ \vdots \\ \boldsymbol{R}_n \end{pmatrix} = \begin{pmatrix} g_1^{r_1} & g_2^{r_1} & \cdots & g_l^{r_1} \\ g_1^{r_2} & g_2^{r_2} & \cdots & g_l^{r_2} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{r_n} & g_2^{r_n} & \cdots & g_l^{r_n} \end{pmatrix}_{n \times l}$
and $Q = (\boldsymbol{Q}_1, \boldsymbol{Q}_2, \cdots, \boldsymbol{Q}_n)^T = \begin{pmatrix} h_1^{r_1} \cdot g^b & h_1^{r_2} & \cdots & h_1^{r_n} \\ h_2^{r_1} & h_2^{r_2} \cdot g^b & \cdots & h_2^{r_n} \\ \vdots & \vdots & \ddots & \vdots \\ h_n^{r_1} & h_n^{r_2} & \cdots & h_n^{r_n} \cdot g^b \end{pmatrix}_{n \times n}$.
When $b = 1$, we say it is in injective mode; otherwise, let $g^0 = 1_G$ and we say it is in lossy mode. At last, the evaluation key is $ek = (R, Q)$.
3. $\mathsf{F}(ek, x)$: Given a message $x = x_1 x_2 \cdots x_n \in \{0, 1\}^n$. Given a function index (R, Q), then calculate $F_{R,Q}(x) = (c_1, c_2)$, where
$c_1 = x \cdot R = (c_{11}, c_{12}, \cdots, c_{1l})$, where $c_{1i} = \prod_{j=1}^n g_i^{r_j x_j}, i \in [l]$;
$c_2 = x \cdot Q = (c_{21}, c_{22}, \cdots, c_{2n})$, where $c_{2i} = \prod_{j=1}^n Q_{ij}^{x_j}, i \in [n]$.
Output $c = (c_1, c_2) \in G^l \times G^n$.
4. $\mathsf{F}^{-1}(td, c)$: Firstly, parse c as $(c_1, c_2) = ((c_{11}, c_{12}, \cdots, c_{1l}), (c_{21}, c_{22}, \cdots, c_{2n}))$.
If $\prod_{j=1}^l c_{1j}^{s_{ij}} = c_{2i}$, then $x_i = 0, i \in [n]$; if $\prod_{j=1}^l c_{1j}^{s_{ij}} \neq c_{2i}$, then $x_i = 1, i \in [n]$.
At last, output the message $x = x_1 x_2 \cdots x_n \in \{0, 1\}^n$.
5. $\mathsf{U}(td, uk)$: Input the update key $uk = \boldsymbol{w}$ and the trapdoor is updated into the new one $td' = td + (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_n) = (\boldsymbol{s}_1 + \boldsymbol{\beta}_1, \boldsymbol{s}_2 + \boldsymbol{\beta}_2, \cdots, \boldsymbol{s}_n + \boldsymbol{\beta}_n)$, where $\boldsymbol{\beta}_i = (b_{i1}, b_{i2}, \cdots, b_{il}) \leftarrow \mathsf{kernel}(\boldsymbol{w})$ (i.e., $s'_{ij} = s_{ij} + b_{ij}$ for $\forall i \in [n], j \in [l]$).

5.2. Correctness and Security

5.2.1. Correctness

- Since the updated trapdoor is $td' = (\boldsymbol{s}_i + \boldsymbol{\beta}_i)_{i\in[n]} = (s_{ij} + b_{ij})_{i\in[n],j\in[l]}$, we have $h'_i = \Pi_{j=1}^{l} g_j^{s_{ij}+b_{ij}} = g^{\langle \boldsymbol{w},\boldsymbol{s}_i+\boldsymbol{\beta}_i\rangle} = g^{\langle \boldsymbol{w},\boldsymbol{s}_i\rangle} = h_i$.
- For any evaluation key ek and $\forall i \in [n]$, there is
  $c_{2i} = \prod_{j=1}^{n} Q_{ij}^{x_j} = g^{bx_i} \cdot \prod_{j=1}^{n} h_i'^{r_j x_j} = g^{bx_i} \cdot h_i'^{\sum_{j=1}^{n} r_j x_j}$
  $= g^{bx_i} \cdot g^{\langle \boldsymbol{w},\boldsymbol{s}_i+\boldsymbol{\beta}_i\rangle \cdot \sum_{j=1}^{n} r_j x_j}$
  $= g^{bx_i} \cdot g^{\langle \boldsymbol{w},\boldsymbol{s}_i\rangle \cdot \sum_{j=1}^{n} r_j x_j}$.
  On the other hand,
  $\prod_{j=1}^{l} c_{1j}^{s'_{ij}} = c_{11}^{s'_{i1}} c_{12}^{s'_{i2}} \cdots c_{1l}^{s'_{il}}$
  $= (\prod_{j=1}^{n} g_1^{r_j x_j})^{s'_{i1}} (\prod_{j=1}^{n} g_2^{r_j x_j})^{s'_{i2}} \cdots (\prod_{j=1}^{n} g_l^{r_j x_j})^{s'_{il}}$
  $= g_1^{s'_{i1} \sum_{j=1}^{n} r_j x_j} g_2^{s'_{i2} \sum_{j=1}^{n} r_j x_j} \cdots g_l^{s'_{il} \sum_{j=1}^{n} r_j x_j}$
  $= g^{w_1 s'_{i1} \sum_{j=1}^{n} r_j x_j} g^{w_2 s'_{i2} \sum_{j=1}^{n} r_j x_j} \cdots g^{w_l s'_{il} \sum_{j=1}^{n} r_j x_j}$
  $= g^{\langle \boldsymbol{w},\boldsymbol{s}_i+\boldsymbol{\beta}_i\rangle \sum_{j=1}^{n} r_j x_j}$
  $= g^{\langle \boldsymbol{w},\boldsymbol{s}_i\rangle \cdot \sum_{j=1}^{n} r_j x_j}$.

  Since in injective mode (i.e., $b = 1$), $g^{bx_i} = g^{x_i}$ holds and the correctness of F and F^{-1} follows.

Theorem 1. *Under the DDH assumption and the (extended) rank hiding assumption in group G with the prime order p, the proposed scheme is a collection of λ-CLR-LTFs with $\lambda \leq (l-2)\log p - 2(\log(1/\epsilon)) + 2$, where $\epsilon = \epsilon(\kappa)$ is some negligible function of the security parameter κ in the floppy model. Therefore, the leakage rate is $\frac{\lambda}{|td|} = \frac{(l-2)\log p - 2(\log(1/\epsilon)+2}{nl\log p} \approx \frac{1}{n}$, and the lossiness is $n - \log p$ bits.*

Proof. Firstly, we prove the lossiness of the proposed scheme is still $n-\log p$ bits after each trapdoor update.

Lossiness. *In the lossy mode, after each trapdoor update, it holds that*

$$h'_i = \Pi_{j=1}^{l} g_j^{s_{ij}+b_{ij}} = g^{\langle \boldsymbol{w},\boldsymbol{s}_i+\boldsymbol{\beta}_i\rangle} = g^{\langle \boldsymbol{w},\boldsymbol{s}_i\rangle} = h_i.$$

Therefore, the evaluation key Q in the lossy mode (i.e., $g^b = 1_G$) is

$$Q = \begin{pmatrix} g^{\langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_1} & g^{\langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_2} & \ldots & g^{\langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_n} \\ g^{\langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_1} & g^{\langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_2} & \ldots & g^{\langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_n} \\ \vdots & \vdots & \ddots & \vdots \\ g^{\langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_1} & g^{\langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_2} & \ldots & g^{\langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_n} \end{pmatrix}_{n\times n}$$

$$= g^{\begin{pmatrix} \langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_1 & \langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_2 & \cdots & \langle \boldsymbol{w},\boldsymbol{s_1}\rangle \cdot r_n \\ \langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_1 & \langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_2 & \cdots & \langle \boldsymbol{w},\boldsymbol{s_2}\rangle \cdot r_n \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_1 & \langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_2 & \cdots & \langle \boldsymbol{w},\boldsymbol{s_n}\rangle \cdot r_n \end{pmatrix}_{n\times n}} = g^{Q'}.$$

Hence, Q' is a matrix of rank 1 since the i-th column is r_i/r_1 times of the first column for all $i \in [n]$ and $i \neq 1$. Therefore, the image of F has size at most $\log p$. The lossiness is $n-\log p$ bits.

In the following, we give the lemma to support the theorem. Based on the λ-CLR-CPA-security of the ElGamal-like public key encryption in the floppy model (Section 4 Lemma 4), the proposed lossy trapdoor function can satisfy the indistinguishability between the injective and lossy functions tolerating at most λ-bit leakage about the trapdoor at each time period, where $\lambda \leq (l-2)\log p - 2\log(1/\epsilon) + 2$.

Lemma 5 (Indistinguishability). *For $\lambda \leq (l-2)\log p - 2(\log(1/\epsilon) + 2$, injective and lossy functions are computationally indistinguishable as long as the leakage number of the trapdoor is less than λ bits between the two trapdoor updates.*

Proof. Let F_{inj} and F_{loss} be the distributions on the injective evaluation key and the lossy evaluation key, respectively. Let F_i be the distribution which is identical to the distribution F_{inj} except for letting the latter i-th main diagonal element $b = 0$ in matrix Q. In evidence, $F_0 = F_{\text{inj}}$, which is the distribution on injective evaluation key and $F_n = F_{\text{loss}}$, which is the distribution on the lossy evaluation key. Therefore, to prove that F_{inj} and F_{loss} are computationally indistinguishable, it is enough to prove that F_{i-1} and F_i are computationally indistinguishable for any $i \in [n]$.

In the following, we show that any distinguisher $\mathcal{D}$ of the two distributions F_{i-1} and F_i can be used to attack the λ-CLR-CPA security of the ElGmal-like PKE scheme. The game is played between a simulator $\mathcal{S}$ and the distinguisher $\mathcal{D}$.

- Given the public key $pk = (G, p, g, g_1, g_2, \cdots, g_l, h)$ of ElGamal-like PKE, the simulator $\mathcal{S}$ chooses a random index $i^* \in [n]$. For $i = [n \backslash i^*]$, the pairs $(\boldsymbol{s}_i, h_i)$ are produced the same as in ElGamal-like PKE. For $i = i^*$, let $h_{i^*} = h$ and $\boldsymbol{s}_i = sk$, where the secret key sk is correlated with the challenge public key pk. Finally, $\mathcal{S}$ sends $pp = (G, p, g, g_1, g_2, \cdots, g_l, h_1, h_2, \cdots, h_n)$ to the distinguisher $\mathcal{D}$.
- Consequently, the simulator $\mathcal{S}$ simulates $\mathcal{D}$'s continuous leakage queries as follows. Suppose that there are polynomial $t = t(\kappa)$ times continuous trapdoor leakage queries. Set $td_0 = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n)$ and $td_i = td_0 + \mathsf{kernal}_i(\boldsymbol{w})(i \in [t])$. We know that the leakage information is a function of $td_i = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n) + \mathsf{kernal}_i(\boldsymbol{w})(i \in [t])$ and the simulator $\mathcal{S}$ knows all $\boldsymbol{s}_i$ except for $\boldsymbol{s}_{i^*}$. According to $\mathcal{D}$'s leakage query function f of $td_i = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n) + \mathsf{kernal}_i(\boldsymbol{w})(i \in [t])$, the simulator $\mathcal{S}$ adapts f as a function of $\boldsymbol{s}_{i^*}$ and presents the function to its own leakage oracles as long as the length of the whole output of f is smaller than λ bits, which is the upper bound of the leakage information of the updatable ElGamal-like PKE scheme. At last, the simulator $\mathcal{S}$ achieves the value $f(td_i)(i \in [t])$ returned from its leakage oracle and then responds with $\mathcal{D}$'s leakage queries.
- The simulator $\mathcal{S}$ simulates the challenge evaluation key as follows. For $(m_0, m_1) = (g, 1_G)$, $\mathcal{S}$ queries its own encryption oracle and gets the challenge ciphertext $C^* = (u_1^*, u_2^*, \cdots, u_l^*, e^*)$, which is the encryption of m_0 or m_1 (i.e., g or 1_G):

  - For $i = [n \backslash i^*]$, choose $\boldsymbol{R}_i = (g_1^{r_i}, g_2^{r_i}, \cdots, g_l^{r_i}) \in L$ with the same witness r_i uniformly at random. Let $\boldsymbol{R}_{i^*} = (u_1^*, u_2^*, \cdots, u_l^*)$ and set $R = (\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_n)^T$.
  - For $i = [n \backslash i^*]$ and $j \in [n]$, compute $h_j^{r_i}$ using the same witness r_i. For $i = i^*, j \in [n]$, let $h_j^{r_i^*} = \Pi_{k=1}^{l} (u_k^*)^{s_{jk}}$ with the secret keys $\boldsymbol{s}_j$.
  - For $i \neq j$, let $Q_{ij} = h_i^{r_j}$. For $1 \leq i \leq i^* - 1$, let $Q_{ii} = h_i^{r_i}$; for $i^* + 1 \leq i \leq n$, let $Q_{ii} = h_i^{r_i} g$; for $i = i^*$, let $Q_{i^* i^*} = e^*$.

The simulator $\mathcal{S}$ sends $ek = (R, Q)$ to $\mathcal{A}$. We can see that when e^* is the encryption of g, the simulator $\mathcal{S}$ simulates a function index based on the distribution F_{i^*-1} perfectly. On the other hand, when e^* is the encryption of 1_G, the simulator $\mathcal{S}$ simulates a function index based on the distribution F_{i^*} perfectly.

At last, the simulator $\mathcal{S}$ outputs what the distinguisher $\mathcal{D}$ outputs. Since $\mathcal{S}$ perfectly simulates F_{i-1}^* or F_{i^*}, according to the challenge ciphertext e^*, for any λ-bit key leakage adversary $\mathcal{D}$, it holds that

$$\Pr[\mathcal{D}(F_{\text{inj}}) = 1] - \Pr[\mathcal{D}(F_{\text{loss}}) = 1] \leq n \cdot \mathrm{Adv}_{\text{ElGamal-like},\mathcal{S}}^{\lambda-\text{CLR-CPA}}(\kappa).$$

Remark 2. *In this section, we can see that the leakage ratio of the DDH-based CLR-LTF is only* $\frac{1}{n}$*, where the lossiness is* $n-logp$*. This relationship implies that the higher the leakage rate, the lower the lossiness. Therefore, it is hard to improve the leakage rate in the prime order group. In the next part, we would like to present an instantiation in the composite order group, which would provide some help in improving the leakage rate to* $1-o(1)$*.*

6. Continuous Leakage Resilient LTFs from the DCR Assumption

In this section, we show how to construct CLR-LTF under the decisional composite residuosity (DCR) assumption. The group $\mathbb{Z}^*_{N^{\alpha+1}}$ is a multiplicative group where $\alpha \geq 1$ is an integer. In addition, the integer $N = PQ$ is an RSA modulus, which means that P and Q are odd primes of equivalent bit length. Obviously, the group $\mathbb{Z}^*_{N^{\alpha+1}}$ is a direct product $\mathsf{G} \times \mathsf{H}$, where G is a cyclic group of order N^α and H is isomorphic to $\mathbb{Z}^*_N$. We define $T := 1 + N(\mathrm{mod} N^{\alpha+1})$; therefore, T generates the group H. In addition, the discrete logarithm with respect to T over group H is efficiently computable. Such an N will be called admissible in the following discussion.

6.1. The Scheme

Set $l = 2 + \frac{\lambda+2\log(1/\epsilon)}{\log N-3}$ for some negligible $\epsilon = \epsilon(\kappa)$. The construction CLR-TDF=(G, S, F, F^{-1}, U) is operated over the group $Z^*_{N^{\alpha+1}}$ as follows.

1. $\mathsf{G}(1^\kappa)$: On inputting 1^κ, the generation algorithm chooses an admissible κ-bit RSA modulus $N = PQ$ and a natural number $\alpha \geq 1$. Note that this fixes the groups G and H (where $g \in \mathsf{G}$ is chosen at random). Set $l(\log N - 2) = \lambda$. Choose $\boldsymbol{s} = (s_1, s_2, \cdots, s_l) \in Z^l_{\frac{N-1}{4}}$ at random. Select $g_1 = g^{w_1}, g_2 = g^{w_2}, \cdots, g_l = g^{w_l} \in \mathsf{G}$ uniformly and let $\boldsymbol{w} = (w_1, w_2, \cdots, w_l) \in \mathbb{Z}^l_{\frac{N-1}{4}}$. Then, $g^{\boldsymbol{w}} = (g_1, g_2, \cdots, g_l)$. Given $h = \Pi^l_{i=1} g_i^{s_i} = g^{\langle \boldsymbol{w}, \boldsymbol{s} \rangle}$, output
$$\mathsf{pp} = (N, \alpha, g, g^{\boldsymbol{w}}, h), td = \boldsymbol{s}, uk = \boldsymbol{w}.$$
2. $\mathsf{S}(\mathsf{pp}, b)$: Given $b \in \{0,1\}$, choose $r \in Z^*_N$ and define
$$R = g^{\boldsymbol{w}r}, Q = h^r \cdot T^b.$$
When $b = 1$, we say it is in injective mode; otherwise, we say it is in lossy mode. At last, the evaluation key is $ek = (R, Q) \in \mathsf{G}^l \times Z^*_{N^{\alpha+1}}$.
3. $\mathsf{F}(ek, x)$: Given a message $x \in Z_{N^\alpha}$. Given a function index (R, Q), then calculate $F_{R,Q}(x) = (c_1, c_2)$, where
$$c_1 = x \cdot R = R^x; \;\; c_2 = x \cdot Q = Q^x.$$
Output $c = (c_1, c_2) \in \mathsf{G}^l \times Z^*_{N^{\alpha+1}}$.
4. $\mathsf{F}^{-1}(td, c)$: Firstly, parse c as (c_1, c_2). In the injective mode, we compute $X = c_2 \cdot (c_1^{-\boldsymbol{s}}) = T^x$. At last, output the message $x = \log_T X$.
5. $\mathsf{U}(td, uk)$: Given the update key $uk = \boldsymbol{w}$ and the trapdoor is updated into the new one $td' = td + \boldsymbol{\beta} = \boldsymbol{s} + \boldsymbol{\beta}$, where $\boldsymbol{\beta} \leftarrow \mathsf{kernel}(\boldsymbol{w})$.

6.2. Correctness and Security

6.2.1. Correctness

- Since the updated trapdoor is $td' = \boldsymbol{s} + \boldsymbol{\beta}$, we have $h' = g^{\langle \boldsymbol{w}, \boldsymbol{s}+\boldsymbol{\beta} \rangle} = g^{\langle \boldsymbol{w}, \boldsymbol{s} \rangle} = h$.
- For any evaluation key ek, there exist
$$c_2 \cdot (c_1^{-\boldsymbol{s}}) = Q^x \cdot (R^x)^{-\boldsymbol{s}} = h^{rx} \cdot T^{bx} \cdot (g^{\boldsymbol{w} \cdot rx})^{-\boldsymbol{s}} = h^{rx} \cdot T^{bx} \cdot h^{-rx} = T^{bx},$$

since in injective mode (i.e., $b = 1$), $T^{bx} = T^x$ holds and the correctness of F and F^{-1} follows.

Theorem 2. *If the DDH assumption is hard in* G *and the DCR problem is hard in* $Z^*_{N^{\alpha+1}}$*, then we can construct a collection of* λ*-CLR-TDFs. During each time interval, the proposed scheme can tolerate at most* $\lambda \le (l-2)(\log N - 3) - 2\log(1/\epsilon)$ *bits on the trapdoor, where* $\epsilon = \epsilon(\kappa)$ *is some negligible function with the security parameter* κ*. Therefore, the leakage rate is* $\frac{\lambda}{|td|} = \frac{(l-2)(\log N-3)-2\log(1/\epsilon)}{l(\log N-3)} \approx 1 - o(1)$*. In addition, the lossiness is at least* $\alpha log N - (log N - 2)$ *bits.*

Proof. Firstly, we prove the lossiness of the proposed scheme is still $\alpha \log N - (\log N - 2)$ bits even after any trapdoor update.

Lossiness. *After each trapdoor update,* $h' = g^{\langle w, s+\beta \rangle} = g^{\langle w, s \rangle} = h$*. Therefore, in the lossy mode (i.e.,* $b = 0$*), it holds* $c_2 = x \cdot Q = h^{rx} \cdot T^{bx} = h^{rx} \in G$ *for any* $x \in Z_{N^\alpha}$*. The image of* F *has size at most* $|G|$*. Since* $N/8 \le |G| \le N/4$*, the lossiness is at least* $\alpha log N - (log N - 2)$ *bits.*

6.2.2. Leakage Rate

Since $N/8 \le |\mathsf{G}| \le N/4$, the min-entropy of the trapdoor is at least $l(\log N - 3)$ bits. The entropy information about the trapdoor revealed by the public key h is at most $\log N - 2$ bits. According to the leftover hash lemma [27]

$$\tilde{H}_\infty(td|(\mathsf{pp}, \lambda)) \ge l(\log N - 3) - (\log N - 2) - \lambda \ge (\log N - 2) + 2\log(1/\epsilon) - 2.$$

Therefore, it holds that $\lambda \le (l-2)(\log N - 3) - 2\log(1/\epsilon)$. As a result, the leakage rate is $\frac{\lambda}{|td|} = \frac{(l-2)(\log N-3)-2\log(1/\epsilon)}{l(\log N-3)} \approx 1 - o(1)$. Clearly, the leakage rate would arrive at 1 with the parameter l increasing.

Lemma 6. *Under the assumption that the DDH assumption is hard in* G *and the DCR problem is hard in* $Z^*_{N^{\alpha+1}}$*, if the extended rank hiding assumption holds, then the scheme implies a* λ*-CLR-CPA secure PKE scheme as long as the leakage parameter* $\lambda \le (l-2)(log N - 3) - 2log(1/\epsilon)$*, where* $\epsilon = \epsilon(\kappa)$ *is some negligible function about* κ*.*

Proof. Obviously, we can extract a DCR-Based ElGamal-like PKE scheme against continuous leakage from the proposed scheme where we can replace the variant b with a message m. As a result, the evaluation key (R, Q) is just the ciphertext of the message m. It is clear that the DCR assumption is properly embedded into the ElGamal-like PKE scheme. Therefore, with the assumption of the DDH and DCR assumptions holding in group G and in $Z^*_{N^{\alpha+1}}$, respectively, and with the extended rank hiding assumption, the result scheme is a CLR-CPA secure PKE scheme with the leakage parameter $\lambda \le (l-2)(\log N - 3) - 2\log(1/\epsilon)$.

According to this lemma, it is natural to reduce the following lemma about the indistinguishability of the injective and lossy function.

Lemma 7 (Indistinguishability). *Under the assumption that the DDH assumption is hard in* G *and the DCR problem is hard in* $Z^*_{N^{\alpha+1}}$*, if the extended rank hiding assumption holds, then the injective and lossy functions are still computationally indistinguishable from the continuous leakage as long as the leakage number of the trapdoor is less than* λ *bits, where* $\lambda \le (l-2)(log N - 3) - 2log(1/\epsilon)$*, and where* $\epsilon = \epsilon(\kappa)$ *is some negligible function about* κ*.*

7. Conclusions

In this paper, we focus on the lossy trapdoor functions in the presence of continuous leakage. Firstly, we introduce the new notion of updatable lossy trapdoor functions and give the formal definition and security requirements. Meanwhile, we extend the notion of ULTFs to CLR-LTFs and give the explicit security model of CLR-LTFs. Then, we introduce the security properties of the CLR ElGamal-like PKE scheme, which will be embedded into our proposed scheme. Under the standard DDH assumption and DCR assumption, respectively, we introduce two concrete lossy trapdoor functions against continuous leakage in the standard model. In these schemes, the trapdoor can be refreshed at regular intervals and the adversaries can learn unbounded leakage information on the trapdoor along the whole system life. Even though, the proposed CLR-LTFs can also be indistinguishable between the injective and lossy evaluation keys. On the other hand, we also show the performance of the proposed schemes compared with the known existing CLR-LTFs. In form, our proposed scheme can also be seen as a deterministic public key encryption, and we think it is of independent interest in the study of efficient deterministic PKE against continuous leakage.

Acknowledgments: This work is the extended version of the paper [28] which has been published in the proceeding of ProvSec'16. And it was partly supported by the National Natural Science Foundation of China under Grant Nos. 61370224, 61672289 and 61672010, the Jiangsu Government Scholarship for Overseas Studies, the Postdoctoral Science Foundation of Jiangsu Province (No. 1601008A), the Natural Science Fund for Colleges and Universities of Jiangsu Province (No. 16KJB520018) and the Youth Cultivation Fund of Nanjing Tech University under ZKJ201528. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

Author Contributions: Sujuan Li conceived and designed the new definition of updatable lossy trapdoor functions and the two main protocols about the continuous leakage resilient lossy trapdoor functions; Sujuan Li, Mingwu Zhang, Yi Mu and Futai Zhang analyzed the security proof of the two theorems: Theorem 1 and Theorem 2; and Sujuan Li wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Peikert, C.; Waters, B. Lossy trapdoor functions and their applications. In Proceedings of the 40th ACM Symposium on Theory of Computing (STOC 2008), Victoria, BC, Canada, 17–20 May 2008; pp. 187–196.
2. Wee, H. Dual projective hashing and its applications–lossy trapdoor functions and more. In *Advances in EUROCRYPT 2012.*; Springer: Berlin/Heidelberg, Germnay, 2012; pp. 246-262.
3. Boldyreva, A.; Fehr, S.; O'Neill, A. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology—CRYPTO 2008, Proceedings of the 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2008*; Wagner, D., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5157, pp. 335–359.
4. Bellare, M.; Brakerski, Z.; Naor, M.; Ristenpart, T.; Segev, G.; Shacham, H.; Yilek, S. Hedged public-key encryption: How to protect against bad randomness. In *Advances in Cryptology ASIACRYPT 2009*; Springer: Berlin/Heidelberg, Germnay, 2009 ; Volume 5912, pp. 232–249.
5. Bellare, M.; Hofheinz, D.; Yilek, S. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Advances in Cryptology—EUROCRYPT 2009, Proceedings of the 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, 26–30 April 2009*; Joux, A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5479; pp. 1–35.
6. Nishimaki, R.; Fujisaki, E.; Tanaka, K. Efficient non-interactive universally composable string-commitment schemes. In *Provable Security, Proceedings of the Third International Conference on Provable Security, Guangzhou, China, 11–13 November 2009*; Pieprzyk, J., Zhang, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5848, pp. 3–18.
7. Halderman J.A.; Schoen S.D.; Heninger, N.; Clarkson, W.; Paul, W.; Calandrino J.A.; Feldman A.J.; Appelbaum, J.; Felten, E.W. Lest we remember: Cold boot attacks on encryption keys. In Proceedings of the 17th USENIX Security Symposium, 28 July–1 August 2008, San Jose, CA, USA; pp. 45–60.
8. Naor, M.; Segev, G. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology–CRYPTO 2009, Proceedings of the 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2009*; Halevi, S., Ed.; Lecture Notes in Computer Science; Springer: Heidelberg, Germnany, 2009; Volume 5677, pp. 18–35.

9. Alwen, J.; Dodis, Y.; Naor, M.; Segev, G.; Walfish, S.; Wichs, D. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology–EUROCRYPT 2010, Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Riviera, French, 30 May–3 June 2010*; Gilbert, H., Ed.; Lecture Notes in Computer Science; Springer: Heidelberg, Germnany, 2010; Volume 6110, pp. 113–134.
10. Brakerski, Z.; Goldwasser, S. Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability (or: Quadratic Residuosity Strikes Back). In *Advances in Cryptology–CRYPTO 2010, Proceedings of the 30th Annual Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2010*; Rabin, T., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6223, pp. 1–20.
11. Kiltz, E.; Pietrzak, K. Leakage Resilient ElGamal Encryption. In *Advances in Cryptology–ASIACRYPT 2010, Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, 5–9 December 2010*; Abe, M., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6477, pp. 595–612.
12. Akavia, A.; Goldwasser, S.; Vaikuntanathan, V. Simultaneous hardcore bits and cryptography against memory attacks. In Proceedings of the 6th Theory of Cryptography, San Francisco, CA, USA, 15–17 March 2009; pp. 474–495.
13. Li, S.; Zhang, F. Leakage-resilient identity-based encryption scheme. *Int. J. Grid Util. Comput.* **2013**, *4*, 187–196.
14. Li, S.; Zhang, F.; Sun, Y.; Shen, L. Efficient leakage resilient public key encryption from DDH assumption. *Cluster Comput.* **2013**, *16*, 797–806.
15. Dodis, Y.; Haralambiev, K.; Lpez-Alt, A.; Wichs, D. Cryptography against continuous memory attacks. In Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010), Las Vegas, NV, USA, 23–26 October 2010; pp. 511–520.
16. Brakerski, Z.; Kalai Y.T.; Katz, J.; Vaikuntanathan, V. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010), Las Vegas, NV, USA, 23–26 October 2010; pp. 501–510.
17. Agrawal, S.; Dodis, Y.; Vaikuntanathan, V.; Wichs, D. On continual leakage of discrete log representations. In *Advances in Cryptology—ASIACRYPT 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 401–420.
18. Yang, R.; Xu, Q.; Zhou, Y.; Zhang, R.; Hu, C.; Yu, Z. Updatable Hash Proof System and Its Applications. In Proceedings of the European Symposium on Research in Computer Security (ESORICS) 2015, Vienna, Austria, 23–25 September 2015; pp. 266–285.
19. Lewko A.B., Rouselakis, Y.; Waters, B. Achieving leakage resilience through dual system encryption. In Proceedings of the Eighth IACR Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, 28–30 March 2011; pp. 70–88.
20. Boyle, E.; Goldwasser, S.; Jain, A.; Kalai Y.T. Multiparty computation secure against continual memory leakage. In Proceedings of the 44th ACM Symposium on Theory of Computing (STOC 2012), New York, NY, USA, 19–22 May 2012; pp. 1235–1254.
21. Ananth, P.; Goyal, V.; Pandey, O. Interactive proofs under continual memory leakage. In Proceedings of the 34th International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2014; pp. 164–182.
22. Alwen, J.; Dodis, Y.; Wichs, D. Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In Proceedings of the 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2009; pp. 36–54.
23. Koppula, V.; Pandey, O.; Rouselakis, Y.; Waters, B. Deterministic Public-Key Encryption under Continual Leakage. In Proceedings of the 14th International Conference on Applied Cryptography and Network Security, ACNS 2016, Guildford, UK, 19–22 June 2016; pp. 304–323.
24. Qin, B.; Liu, S.; Chen, K.; Charlemagne, M. Leakage-resilient lossy trapdoor functions and public-key encryption. In Proceedings of the 2013 ACM Asia Public-Key Cryptography Workshop, Hangzhou, China, 8 May 2013.; pp. 3–12.
25. Boneh, D.; Halevi, S.; Hamburg, M.; Ostrovsky R. Circular-Secure Encryption from Decision Diffie–Hellman. In Proceedings of the 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2008; pp. 108–125.
26. Naor, M.; Segev, G. Public-Key Cryptosystems Resilient to Key Leakage. In *Advances in Cryptology—CRYPTO'09*; Springer: Berlin/Heidelberg, Germnay, 2009; pp. 18–35.

27. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139.
28. Li, S.; Mu, Y.; Zhang, M.; Zhang, F. Updatable Lossy Trapdoor Functions and Its Application in Continuous Leakage. In Proceedings of the 10th International Conference on Provable Security (ProvSec 2016), Nanjing, China, 10–12 November 2016; pp. 309–319.