General Construction of Chameleon All-But-One Trapdoor Functions and Their Applications

Jinyong Chang[a,b], Rui Xue[a]
[a] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Email: {changjinyong,xuerui}@iie.ac.cn
[b] Department of Mathematics, Changzhi University, Changzhi, Shanxi 046011, China

Abstract— **Chameleon all-but-one trapdoor function (ABO-TDF) is an important and useful primitive which was introduced in [9]. With the help of it, a more efficient black-box construction of public key encryption (PKE) scheme, which is secure against chosen-ciphertext attack (CCA), can be given. In this paper, we formally generalize the construction of chameleon ABO-TDFs. As a special case of our generalization, a concrete construction of ABO-TDFs, which was first introduced by Peikert and Waters [1], is presented. Although the existence of lossy trapdoor functions is equivalent to that of ABO-TDFs by using the conversion in [1], as Peikert et al. said, the conversion involves some degradation in lossiness (i.e. additional leakage). Therefore, in this sense, our result is different from those in [21] where Hemenway et al. proved that homomorphic encryption with some additional properties implies lossy trapdoor functions.**

Index Terms— **lossy trapdoor functions; chameleon all-but-one trapdoor functions; chosen ciphertext security; homomorphic encryption**

I. INTRODUCTION

LOSSY trapdoor functions (LTDFs) were first introduced by Peikert and Waters [1]–[3] and further studied in [4]–[12]. LTDFs imply lots of fundamental cryptographic primitives, such as collision-resistant hash functions, oblivious transfer, etc. In addition, they can also be used to construct many cryptographic schemes, such as deterministic public-key encryption, encryption and commitments that are secure against selective opening attacks. Most important of all, LTDFs enable black-box construction of CCA secure PKE schemes (see [13]–[20] and their references).

LTDF is centered around the idea of losing information. Informally, LTDF is a public function f that is created to behave in one of two ways. The first way corresponds to the usual completeness condition for an (injective) trapdoor function: given a suitable trapdoor for f, the entire input x can be efficiently recovered from $f(x)$. In the second way, f statistically loses a significant amount of information about its input, i.e. most outputs of f have many preimages. Finally, the two behaviors are indistinguishable: given just the public description of f, no efficient adversary can tell whether f is injective or lossy.

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

The black-box construction of CCA-secure PKE from LTDFs in [1] needs a collection of LTDFs, a collection of ABO-TDFs, a pair-wise independent family of hash functions, and a strongly unforgeable one-time signature scheme, where the set of verification keys is a subset of the branch set of the ABO collection.

Since LTDFs implies CCA secure PKE schemes, in [21], Hemenway and Ostrovsky considered the problem whether homomorphic encryption implies LTDFs. Fortunately, they found the "bridge" and presented an excellent construction of LTDFs from homomorphic encryption schemes with some additional properties.

About LTDFs and ABO-TDFs, in [1], Peikert and Waters has proved that, from the perspective of existence, LTDFs are equivalent to ABO-TDFs with binary branch sets and an ABO collection for large branch sets can also be constructed from one with just binary branch set. However, their conversion involves some degradation in lossiness (i.e. additional leakage) and whether this can be improved remains open [1].

Lai et al. [9] introduced a new notion named chameleon ABO-TDFs whose goal is to replace ABO-TDFs and strongly unforgeable one-time signature in the construction of CCA-secure PKE schemes, which can improve the efficiency of [1]. They also proposed a concrete construction of chameleon ABO-TDFs based on any secure against chosen plaintext attack (CPA) homomorphic public key encryption scheme with some additional properties, like the Damgård-Jurik encryption scheme [22].

Manuscript received December 13, 2013; revised April 17, 2014; accepted April 18, 2014. © 2005 IEEE.
This work is supported by National Natural Science Foundation of China (No.61170280), the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDA06010701), and the Foundation of Institute of Information Engineering for Cryptography.

doi:10.4304/jsw.9.10.2557-2563

A. Our Contributions

In this paper, we formally generalize the construction of chameleon ABO-TDFs, which includes their construction as a special case. As another special case of our generalization, a concrete construction of ABO-TDFs instead of lossy trapdoor functions is also presented, which does not involve degradation of lossiness. Since the lossiness amount of information is of key importance for the construction of CCA-secure public key schemes, in this sense, our result is different from that of [21].

B. Organization of the Paper

The remainder of the paper is organized as follows. In Section 2 we review some standard notions and cryptographic definitions. In Section 3 we describe our generic construction of chameleon ABO-TDFs. In Section 4, we give two special cases which can be regarded as two applications of our generic construction. In Section 5, we describe a simple homomorphic encryption scheme which can be used in our construction.

II. Preliminaries

In this section we review some standard notations and cryptographic definitions.

A. Basic Concepts

Let $\mathbb{N}$ be the set of natural numbers. If M is a set, then we let $|M|$ denote its size and $m \xleftarrow{R} M$ denote picking element m uniformly at random from M. Let λ be a security parameter and PPT is probabilistic polynomial time. Let $z \leftarrow A(x, y, \cdots)$ denote the operation of running an algorithm A with inputs $(x, y, \cdots)$ and output z. A function $\mathrm{negl}(\lambda)$ is *negligible* (in λ) if $\mathrm{negl}(\lambda) = o(\lambda^{-c})$ for any constant $c > 0$. Let U_ℓ denote the uniform distribution on ℓ-bit binary strings.

B. Cryptographic Notions

1) Lossy Trapdoor Functions: Let $n(\lambda) = \mathrm{poly}(\lambda)$ be the length of the functions's input and $k(\lambda) \leq n(\lambda)$ denotes the *lossiness* of the collection. Moreover, we also define the *residual leakage* value $r(\lambda) = n(\lambda) - k(\lambda)$.

Definition 1 (Lossy Trapdoor Functions [1]–[3]): A collection of (n, k)-lossy trapdoor functions is given by a tuple of probabilistic polynomial-time algorithms $(S_{\mathrm{inj}}, S_{\mathrm{loss}}, F, F^{-1})$ having the following properties.

1) Easy to sample an injective function with trapdoor: $S_{\mathrm{inj}}(1^\lambda)$ 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) Easy to sample a lossy function: $S_{\mathrm{loss}}(1^\lambda)$ outputs s, where s is a function index, and $F(s, \cdot)$ 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 ensembles $\{s : (s, t) \leftarrow S_{\mathrm{inj}}(1^\lambda)\}_{\lambda\in\mathbb{N}}$ and $\{s : s \leftarrow S_{\mathrm{loss}}(1^\lambda)\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable.

Remark 2: In fact, we also consider a relaxed definition of lossy TDFs in some situations, which we call *almost-always* lossy TDFs. In particular, we need that with overwhelming probability, the description s of a function describes an injective function while $F^{-1}(s, \cdot)$ is the inversion of $F(s, \cdot)$. Moreover, we also require that the lossy function $F(s, \cdot)$ generated from S_{loss} has image size at most $2^r = 2^{n-k}$ with overwhelming probability .

2) All-But-One Lossy Trapdoor Functions: The new notion of ABO-TDFs is a richer abstraction of LTDFs. Briefly, in an ABO collection, every function has another extra input which is called its branch. All the branches are injective TDFs, except for one branch that is lossy. Formally,

Definition 3 (ABO-TDFs [1]–[3]): A collection of (n, k)-all-but-one trapdoor functions with branch collection $\mathbb{B}$ is given by a tuple of possibly PPT algorithms $(S_{\mathrm{abo}}, G_{\mathrm{abo}}, G_{\mathrm{abo}}^{-1})$ having the following properties:

1) Sampling a trapdoor function with given lossy branch: for any $b^* \in B_\lambda$, $S_{\mathrm{abo}}(b^*)$ outputs (s, t), where s is a function index and t is its trapdoor.
2) Evaluation of injective functions: For any $b \in B_\lambda$ distinct from b^*, $G_{\mathrm{abo}}(s, b, \cdot)$ computes an injective (deterministic) function $g_{s,b}(\cdot)$ over the domain $\{0, 1\}^n$, and $G_{\mathrm{abo}}^{-1}(t, b, \cdot)$ computes $g_{s,b}^{-1}(\cdot)$.
3) Evaluation of lossy functions: $G_{\mathrm{abo}}(s, b^*, \cdot)$ computes a function $g_{s,b^*}(\cdot)$ over the domain $\{0, 1\}^n$ whose image has size at most $2^r = 2^{n-k}$.
4) Hidden lossy branch: The ensembles $\{s : (s, t) \leftarrow S_{\mathrm{abo}}(b_0^*)\}_{\lambda\in\mathbb{N}, b_0^*\in B_\lambda}$ and $\{s : (s, t) \leftarrow S_{\mathrm{abo}}(b_1^*)\}_{\lambda\in\mathbb{N}, b_1^*\in B_\lambda}$ are computationally indistinguishable.

Remark 4: Just as lossy TDFs, we also consider a relaxed definition of ABO-TDFs and we call it *almost-always* ABO-LTDFs. In particular, we require that the injective, invertible and lossy properties hold with overwhelming probability.

3) Chameleon All-But-One Trapdoor Functions: Chameleon ABO-TDFs is also a specific kind of ABO-TDFs with two variable (a, b) as a branch [9]. We recall it as follows. Let $\mathbb{A}\times\mathbb{B} = \{A_\lambda \times B_\lambda\}_{\lambda\in\mathbb{N}}$ be two families of sets whose elements is the branches. Then,

Definition 5 (Chameleon ABO-TDFs [9], [10]): A family of (n, k)-chameleon ABO-LTDFs consists of four PPT algorithms $(S_{\mathrm{ch}}, F_{\mathrm{ch}}, F_{\mathrm{ch}}^{-1}, \mathrm{CLB}_{\mathrm{ch}})$ which have the following properties:

1) Sampling a function: For any $\lambda \in \mathbb{N}$, $S_{\mathrm{ch}}(1^\lambda)$ outputs (s, t, Q) where s is a function index, t is its trapdoor and $Q \subset A_\lambda \times B_\lambda$ is a set of lossy branches.
2) Evaluation of injective functions: For any $(a, b) \in A_\lambda \times B_\lambda$, if $(a, b) \notin Q$ where $(s, t, Q) \leftarrow S_{\mathrm{ch}}(1^\lambda)$, then $F_{\mathrm{ch}}(s,\ a, b, \cdot)$ computes a injective function $g_{s,a,b}(\cdot)$ over the domain $\{0, 1\}^n$, and $F_{\mathrm{ch}}^{-1}(s, t, a, b, \cdot)$ computes $g_{s,a,b}^{-1}(\cdot)$.

3) Evaluation of lossy functions: For any $(a, b) \in A_\lambda \times B_\lambda$, if $(a, b) \in Q$ where $(s, t, Q) \leftarrow S_{\rm ch}(1^\lambda)$, then $F_{\rm ch}(s, a, b, \cdot)$ computes a function $g_{s,a,b}(\cdot)$ over the domain $\{0, 1\}^n$ whose image has size at most 2^{n-k}.
4) Chameleon property: On input the function index s, the trapdoor t and any $a \in A_\lambda$, $\mathrm{CLB_{ch}}$ computes a unique $b \in B_\lambda$ to result in a lossy branch (a, b). In formula, $b \leftarrow \mathrm{CLB_{ch}}(s, t, a)$ such that $(a, b) \in Q$.
5) Security (1)—Indistinguishability between lossy branches and injective branches: It is hard to distinguish a lossy branch from an injective branch. Any PPT algorithm $\mathcal{A}$ that receives s as input, where $(s, t, Q) \leftarrow S_{\rm ch}(1^\lambda)$, has only a negligible probability of distinguishing a pair $(a, b_0) \in Q$ from $(a, b_1) \notin Q$, even a is chosen by $\mathcal{A}$. Formally, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a distinguisher *CH-LI* and define its advantage as $\mathrm{Adv}_{\mathcal{A}}^{\textit{CH-LI}}(1^\lambda)$:

$$\left| \Pr\left[\beta = \beta' : \begin{array}{c} (s, t, Q) \leftarrow S_{\rm ch}(1^\lambda); \\ a \leftarrow \mathcal{A}_1(s); \\ b_0 = \mathrm{CLB_{ch}}(s, t, a); \\ b_1 \xleftarrow{R} B_\lambda; \beta \xleftarrow{R} \{0, 1\}; \\ \beta' \leftarrow \mathcal{A}_2(s, a, b_\beta) \end{array} \right] - \frac{1}{2} \right|.$$

It is hard to distinguish a lossy branch from an injective branch, if $\mathrm{Adv}_{\mathcal{A}}^{\textit{CH-LI}}(\cdot)$ is negligible for every PPT distinguisher $\mathcal{A}$.
6) Security (2)—Hard to find one-more lossy branch: Any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives (s, a, b) as input, where $(s, t, Q) \leftarrow S_{\rm ch}(1^\lambda)$ and $(a, b) \xleftarrow{R} Q$, has at most a negligible probability of outputting a pair $(a', b') \in Q \backslash \{(a, b)\}$.

If $F_{\rm ch}^{-1}(t, a, b, \cdot)$ can correctly invert all images of $g_{s,a,b}(\cdot)$ with $(a, b) \notin Q$ and $\mathrm{CLB_{ch}}(s, t, a)$ output b satisfying $(a, b) \in Q$, both with *overwhelming probability*, then we call the collection *almost-always* chameleon ABO-TDFs.

4) Homomorphic Encryption:

Definition 6 (Homomorphic Encryption [9], [10]): A PKE scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is called homomorphic if:

- It is CPA seucre;
- The plaintext space is a group M and we denote the group operation by "+";
- all the ciphertexts are members of a group C';
- $\forall x_0, x_1 \in X$, and $\forall r_0, r_1$ in R, there exists an $r^* \in R$ satisfying

$$\mathrm{Enc}_{pk}(x_0 + x_1, r^*) = \mathrm{Enc}_{pk}(x_0, r_0)\mathrm{Enc}_{pk}(x_1, r_1).$$

III. Generic Construction of Chameleon ABO-TDFs

Let $d = d(\lambda), k = k(\lambda), l = l(\lambda)$ be three polynomials of λ. Let $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ be a homomorphic encryption scheme with plaintext space M and randomness space R satisfying $|M| > d|R|$. In addition, we assume that

1) M is a finite field (or commutative ring with multiplicative identity and, with *overwhelming probability*, each element in M has multiplicative inverse when we construct *almost-always* chameleon ABO-TDFs).
2) For $a, m \in M$, $(\mathrm{Enc}_{pk}(m))^a = \mathrm{Enc}_{pk}(am)$.

In the following, we give the description of our chameleon ABO-TDFs $(S_{\rm ch}, F_{\rm ch}, F_{\rm ch}^{-1}, \mathrm{CLB_{ch}})$ with the set of branches $\mathbb{A} \times \mathbb{B} = \{A_\lambda \times B_\lambda\}_{\lambda \in \mathbb{N}} = \{M^k \times M^l\}_{\lambda \in \mathbb{N}}$:

- Sampling a function: $S_{\rm ch}$ takes as input 1^λ. It invokes $\mathrm{Gen}(1^\lambda)$ to generate (pk, sk). Then it samples uniformly at random a matrix $D = (A_{d\times(k+1)}, B_{d\times l}) = (x_{ij})_{d\times(k+l+1)}$ satisfying $rank(B_{d\times l}) \geq l$, where $x_{ij} \in M$ for $1 \leq i \leq d, 1 \leq j \leq k + l + 1$, and computes $C = (c_{ij})_{d\times(k+l+1)} = (\mathrm{Enc}_{pk}(x_{ij}))_{d\times(k+l+1)}$. It outputs the function index $s = (pk, C)$, the trapdoor $t = (sk, D)$ and the set of lossy branches Q which is the set of all pairs of $(a_1, \cdots, a_k, b_1, \cdots, b_l) \in M^{k+l}$ satisfying
$$\left\{ \begin{array}{l} a_1x_{11} + \cdots + a_kx_{1k} + x_{1,k+1} \\ \qquad +b_1x_{1,k+2} + \cdots + b_lx_{1,k+l+1} = 0, \\ a_1x_{21} + \cdots + a_kx_{2k} + x_{2,k+1} \\ \qquad +b_1x_{2,k+2} + \cdots + b_lx_{2,k+l+1} = 0, \\ \quad\cdots \\ a_1x_{d1} + \cdots + a_kx_{dk} + x_{d,k+1} \\ \qquad +b_1x_{d,k+2} + \cdots + b_lx_{d,k+l+1} = 0. \end{array} \right. \tag{1}$$
- Evaluating a function: $F_{\rm ch}$ takes as input $(s, a_1, \cdots a_k, \ b_1, \cdots b_l, x)$, where $x \in M$. It computes
$$\begin{array}{l} y = \big((c_{11}^{a_1} \cdots c_{1k}^{a_k} c_{1,k+1} c_{1,k+2}^{b_1} c_{1,k+l+1}^{b_l})^x, \\ \qquad (c_{21}^{a_1} \cdots c_{2k}^{a_k} c_{2,k+1} c_{2,k+2}^{b_1} \cdots c_{2,k+l+1}^{b_l})^x, \\ \qquad \cdots, (c_{d1}^{a_1} \cdots c_{dk}^{a_k} c_{d,k+1} c_{d,k+2}^{b_1} \cdots c_{d,k+l+1}^{b_l})^x\big) \\ := (y_1, \cdots, y_d) \end{array}$$
and outputs y.
- Inverting an injective function: $F_{\rm ch}^{-1}$ takes as input $(s, t, a_1, \cdots a_k, b_1, \cdots b_l, y)$, where $(a_1, \cdots, a_k, b_1, \cdots, b_l) \notin Q$. Choose $i \in \{1, \cdots, d\}$ satisfying
$$\begin{array}{l} a_1x_{i1} + \cdots + a_kx_{ik} + x_{i,k+1} \\ \qquad + b_1x_{i,k+2} + \cdots + b_lx_{i,k+l+1} \neq 0 \end{array}$$
and outputs
$$\begin{array}{l} x = \mathrm{Dec}_{sk}(y_i) \cdot \big(a_1x_{i1} + \cdots + a_kx_{ik} + x_{i,k+1} \\ \qquad +b_1x_{i,k+2} + \cdots + b_lx_{i,k+l+1}\big)^{-1} \end{array}$$
- Computing a lossy branch: $\mathrm{CLB_{ch}}$ takes as input $(s, t, \ a_1, \cdots, a_k)$. It can obtain $(b_1, \cdots, b_l)$ since $rank(B_{d\times l}) \geq l$, where $B_{d\times l}$ is the matrix that was sampled to satisfying (1). Then it output $(b_1, \cdots, b_l)$.

Now we state our main theorem as follows.

Theorem 7: The algorithms $(S_{\rm ch}, F_{\rm ch}, F_{\rm ch}^{-1}, \mathrm{CLB_{ch}})$ is a collection of $(\log|M|, \ \log|M| - \log d|R|)$-chameleon all-but-one trapdoor functions.

Proof: We prove this theorem from the following aspects:

1) About injective function F_{ch} : For any $(a_1, \cdots, a_k,\ b_1, \cdots, b_l) \in M^{k+l}$, if we know $(a_1, \cdots, a_k, b_1, \cdots, b_l) \notin Q$, then

$$\begin{aligned} y &= F_{\text{ch}}(s, a_1, \cdots a_k, b_1, \cdots b_l, x) \\ &= \big((c_{11}^{a_1} \cdots c_{1k}^{a_k} c_{1,k+1} c_{1,k+2}^{b_1} c_{1,k+l+1}^{b_l})^x, \\ &(c_{21}^{a_1} \cdots c_{2k}^{a_k} c_{2,k+1} c_{2,k+2}^{b_1} \cdots c_{2,k+l+1}^{b_l})^x, \\ &\cdots, (c_{d1}^{a_1} \cdots c_{dk}^{a_k} c_{d,k+1} c_{d,k+2}^{b_1} \cdots c_{d,k+l+1}^{b_l})^x\big) \end{aligned}$$

is a deterministic injective function over the domain M since the randomness of the encryption algorithm has been determined by the relationships of x_{ij} and c_{ij}.

2) About F_{ch}^{-1}: For any $(a_1, \cdots, a_k, b_1, \cdots, b_l) \in M^{k+l}$, if $(a_1, \cdots, a_k, b_1, \cdots, b_l) \notin Q$, then

$$\begin{aligned} y &= F_{\text{ch}}(s, a_1, \cdots a_k, b_1, \cdots b_l, x) \\ &= \big((c_{11}^{a_1} \cdots c_{1k}^{a_k} c_{1,k+1} c_{1,k+2}^{b_1} c_{1,k+l+1}^{b_l})^x, \\ &\quad (c_{21}^{a_1} \cdots c_{2k}^{a_k} c_{2,k+1} c_{2,k+2}^{b_1} \cdots c_{2,k+l+1}^{b_l})^x, \\ &\quad \cdots, (c_{d1}^{a_1} \cdots c_{dk}^{a_k} c_{d,k+1} c_{d,k+2}^{b_1} \cdots c_{d,k+l+1}^{b_l})^x\big) \\ &= \big(([\text{Enc}_{pk}(x_{11})]^{a_1} \cdots [\text{Enc}_{pk}(x_{1k})]^{a_k} \\ &\quad \text{Enc}_{pk}(x_{1,k+1}) \cdot [\text{Enc}_{pk}(x_{1,k+2})]^{b_1} \\ &\quad \cdots [\text{Enc}_{pk}(x_{1,k+l+1})]^{b_l})^x, \\ &\quad ([\text{Enc}_{pk}(x_{21})]^{a_1} \cdots [\text{Enc}_{pk}(x_{2k})]^{a_k} \\ &\quad \text{Enc}_{pk}(x_{2,k+1}) \cdot [\text{Enc}_{pk}(x_{2,k+2})]^{b_1} \\ &\quad \cdots [\text{Enc}_{pk}(x_{2,k+l+1})]^{b_l})^x, \\ &\quad \cdots, \\ &\quad ([\text{Enc}_{pk}(x_{d1})]^{a_1} \cdots [\text{Enc}_{pk}(x_{dk})]^{a_k} \\ &\quad \cdot \text{Enc}_{pk}(x_{d,k+1}) \cdot [\text{Enc}_{pk}(x_{d,k+2})]^{b_1} \\ &\quad \cdots [\text{Enc}_{pk}(x_{d,k+l+1})]^{b_l})^x\big) \\ &= \big(\text{Enc}_{pk}(x(a_1 x_{11} + \cdots + a_k x_{1k} + x_{1,k+1} \\ &\quad + b_1 x_{1,k+2} + \cdots + b_l x_{1,k+l+1})), \\ &\quad \text{Enc}_{pk}(x(a_1 x_{21} + \cdots + a_k x_{1k} + x_{2,k+1} \\ &\quad + b_1 x_{2,k+2} + \cdots + b_l x_{2,k+l+1})), \\ &\quad \text{Enc}_{pk}(x(a_1 x_{d1} + \cdots + a_k x_{dk} + x_{d,k+1} \\ &\quad + b_1 x_{d,k+2} + \cdots + b_l x_{d,k+l+1}))\big). \end{aligned}$$

Since F_{ch}^{-1} chooses $i \in \{1, \cdots, d\}$ satisfying

$$\begin{aligned} &a_1 x_{i1} + \cdots + a_k x_{ik} + x_{i,k+1} \\ &\qquad + b_1 x_{i,k+2} + \cdots + b_l x_{i,k+l+1} \neq 0 \end{aligned}$$

and outputs

$$\begin{aligned} x = \text{Dec}_{sk}(y_i) \cdot \big(&a_1 x_{i1} + \cdots + a_k x_{ik} + x_{i,k+1} \\ &+ b_1 x_{i,k+2} + \cdots + b_l x_{i,k+l+1}\big)^{-1}, \end{aligned}$$

F_{ch}^{-1} can retrieve x correctly.

3) About lossy function F_{ch} : For any $(a_1, \cdots, a_k,$ $b_1, \cdots, b_l) \in M^{k+l}$, if choose $(a_1, \cdots, a_k,$ $b_1, \cdots, b_l) \in Q$, then

$$\begin{aligned} y &= F_{\text{ch}}(s, a_1, \cdots a_k, b_1, \cdots b_l, x) \\ &= \big(\text{Enc}_{pk}(x(a_1 x_{11} + \cdots + a_k x_{1k} + x_{1,k+1} \\ &\qquad + b_1 x_{1,k+2} + \cdots + b_l x_{1,k+l+1})), \\ &\quad \text{Enc}_{pk}(x(a_1 x_{21} + \cdots + a_k x_{1k} + x_{2,k+1} \\ &\qquad + b_1 x_{2,k+2} + \cdots + b_l x_{2,k+l+1})), \\ &\quad \text{Enc}_{pk}(x(a_1 x_{d1} + \cdots + a_k x_{dk} + x_{d,k+1} \\ &\qquad + b_1 x_{d,k+2} + \cdots + b_l x_{d,k+l+1}))\big) \\ &= \big(\text{Enc}_{pk}(0), \text{Enc}_{pk}(0), \cdots, \text{Enc}_{pk}(0)\big). \end{aligned}$$

Then the image size of F_{ch} is at most $d|R|$. Therefore the amount of lossiness is at least $\log|M| - \log d|R|$.

4) Hard to distinguish a lossy branch from an injective branch: Since the scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ is CPA-secure, the scheme $\Pi' = (\text{Gen}', \text{Enc}', \text{Dec}')$, whose encryption algorithm is

$$\text{Enc}'_{pk}(m_1, \cdots, m_d) = (\text{Enc}_{pk}(m_1), \cdots, \text{Enc}_{pk}(m_d))$$

and decryption algorithm is corresponding to it, is also CPA-secure. Now if $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a PPT distinguisher who can distinguish a lossy branch from an injective branch with non-negligible probability. We can use $\mathcal{A}$ to construct a PPT adversary $\mathcal{A}'$ who breaks the CPA-secure of Π'. $\mathcal{A}'$ works as follows:

$\mathcal{A}'$:
on input pk;
samples $x_{11}, \cdots, x_{d,k+l+1} \xleftarrow{R} M$ and $x'_{1,k+1}, \cdots, x'_{d,k+1} \xleftarrow{R} M$;
outputs

$$m_0 = (x_{1,k+1}, \cdots, x_{d,k+1}),$$

and

$$m_1 = (x'_{1,k+1}, \cdots, x'_{d,k+1});$$

receives the challenge ciphertext

$$c^* = (\text{Enc}'_{pk}(m_b))^T;$$

computes $C_b = (c_1, \cdots, c_k, c^*, c_{k+2} \cdots, c_{l+k+1})$, where

$$c_i = (\text{Enc}_{pk}(x_{1i}), \cdots, \text{Enc}_{pk}(x_{di}))^T,$$

for $i \in \{1, \cdots, k, k+2, \cdots, k+l+1\}$;
gives $s = (pk, C_b)$ to $\mathcal{A}$ and obtains $(a_1, \cdots, a_k)$ from $\mathcal{A}$;
computes $(b_1^0, \cdots, b_l^0)$ and $(b_1^1, \cdots, b_l^1)$ using m_0, m_1 and $(a_1, \cdots, a_k)$;
chooses randomly $\beta \in \{0, 1\}$ and gives $(b_1^\beta, \cdots, b_l^\beta)$ to $\mathcal{A}$;
outputs the bit β' which $\mathcal{A}$ outputs.

Obviously, we have

$$\text{Adv}_{\mathcal{A}', \Pi'}^{\text{CPA}}(\lambda) \geq \text{Adv}_{\mathcal{A}}^{\text{CH-LI}}(\lambda).$$

This contradicts the CPA-security of Π'.

5) Hard to find one-more lossy branch: We observe that the matrix $(x_{ij})_{d\times(k+l+1)}$ are hidden by the CPA-secure PKE. $\mathcal{A}$ could obtain the following equations:

$$\begin{cases} a_1x_{11}+\cdots+a_kx_{1k}+x_{1,k+1} \\ \quad +b_1x_{1,k+2}+\cdots+b_lx_{1,k+l+1}=0, \\ a_1x_{21}+\cdots+a_kx_{2k}+x_{2,k+1} \\ \quad +b_1x_{2,k+2}+\cdots+b_lx_{2,k+l+1}=0, \\ \cdots \\ a_1x_{d1}+\cdots+a_kx_{dk}+x_{d,k+1} \\ \quad +b_1x_{d,k+2}+\cdots+b_lx_{d,k+l+1}=0. \end{cases}$$

However, there are $|M|^{k+l}$ kinds of pairs that satisfy those equations and each of them are equally likely. Formally, we prove that if there exists an PPT algorithm $\mathcal{A}$ that can find one-more lossy branch with non-negligible probability, then we can construct PPT inverter $\mathcal{I}$ which can breaks the one-wayness of the PKE scheme. $\mathcal{I}$ works as follows:

$\mathcal{I}$:
on input y, pk;
chooses $x_{ij}\in M$ and computes $c_{ij}=\mathrm{Enc}_{pk}(x_{ij})$ for $1\le i\le d; 1\le j\le k, k+2\le j\le k+l+1$ and additionally chooses $(a_1,\cdots,a_k,b_1,\cdots,b_l)\xleftarrow{R} M$;
computes $c_{i,k+1}=c_{i1}^{a_1}\cdots c_{ik}^{a_k}c_{i,k+2}^{b_1}\cdots c_{i,k+l+1}^{b_l}$, for $1\le i\le d$;
let C be the matrix $(c_{ij})_{d,k+l+1}$; chooses randomly $i_0\in\{1,\cdots,d\}, j_0\in\{1,\cdots,k+l+1\}\backslash\{k+1\}$; let C_1 be the matrix whose element c_{i_0,j_0} is replaced by y (without loss of generality we assume $j_0\le k$);
gives $\big(s=(pk,C_1),(a_1,\cdots,a_k,b_1,\cdots,b_l)\big)$ to $\mathcal{A}$;
if $\mathcal{A}$ outputs $(a_1',\cdots,a_k',b_1',\cdots,b_l')$ and $a_{j_0}\ne a_{j_0}'$, then outputs

$$\begin{aligned} x_{i_0,j_0}=&-(a_{j_0}-a_{j_0}')^{-1}[(a_1-a_1')x_{i_0,1}+\cdots \\ &+((a_{j_0-1}-a_{j_0-1}'))x_{i_0,j_0-1} \\ &\quad+((a_{j_0+1}-a_{j_0+1}'))x_{i_0,j_0+1}+\cdots \\ &+(b_l-b_l')x_{i_0,k+l+1}]. \end{aligned}$$

Obviously, we have

$$\begin{aligned}\Pr[\mathcal{I}\text{ success}]&\ge\frac{1}{(k+l)d}\cdot d\cdot\Pr[\mathcal{A}\text{ success}] \\ &=\frac{1}{(k+l)}\Pr[\mathcal{A}\text{ success}]\end{aligned}$$

This contradicts to the one-wayness of the PKE Π. ■

IV. Two Special Cases

In this section, we give two special cases of our generic construction described above by fixing the corresponding parameters.

A. The First Case

Now we fix the parameters as $d=k=l=1$. We remark that this is the case discussed in [9], where the authors constructed a chameleon all-but-one TDFs and used them to improve efficiency of the CCA scheme presented in [1].

B. The Second Case

In this subsection, we fix the parameters as $d=l=1, k=0$. As previously mentioned, let $\Pi=(\mathrm{Gen},\mathrm{Enc},\mathrm{Dec})$ be a homomorphic encryption scheme with plaintext space M and randomness space R satisfying $|M|>|R|$. In addition, we assume that

1) M is a finite field (or commutative ring with multiplicative identity and, with *overwhelming probability*, each element in M has multiplicative inverse when we construct *almost-always* ABO-TDFs)
2) For $a,m\in M$, $(\mathrm{Enc}_{pk}(m))^a=\mathrm{Enc}_{\mathrm{pk}}(am)$.

We remark that, in this situation, we essentially give a construction of ABO-TDFs with branches set M. In the following, we describe the construction in detail. A collection of ABO-TDFs $(S_{\mathrm{abo}},G_{\mathrm{abo}},G_{\mathrm{abo}}^{-1})$ is defined as follows:

- $S_{\mathrm{abo}}(1^\lambda,b^*)$: It invokes $\mathrm{Gen}(1^\lambda)$ to generate (pk,sk). Then it samples randomly $x_1,x_2\in M$ satisfying $b^*x_1+x_2=0$ and computes $c_1=\mathrm{Enc}_{pk}(x_1), c_2=\mathrm{Enc}_{pk}(x_2)$. The function index is $s=(pk,c_1,c_2)$ and the trapdoor is $t=(sk,x_1,x_2)$.
- Evaluating a function G_{abo}: takes as input (s,b,x), where $(s,t)\leftarrow\mathrm{Gen}(1^\lambda)$, $b,x\in M$, it computes $y=G_{\mathrm{abo}}(s,b,x)=(c_1^bc_2)^x$ and outputs y.
- Inverting an injective function G_{abo}^{-1} : takes as input (t,b,y), where $b\ne b^*$, it computes

$$x=\mathrm{Dec}_{sk}(y)(bx_1+x_2)^{-1}.$$

Theorem 8: The algorithms described above give a collection of $(\log|M|,\log|M|-\log|R|)$ almost-always all-but-one trapdoor functions.

Proof: 1. For any $b\in M$ distinct from b^*, $G_{\mathrm{abo}}(s,b,x)=(c_1^bc_2)^x$ computes an deterministic function over the domain M since the randomness of $\mathrm{Enc}_{pk}(\cdot)$ has been determined by the relationship between x_1,x_2 and c_1,c_2. Meanwhile, the function $G_{\mathrm{abo}}(s,b,x)$ is always injective except that the event that $x_1=0$ whose probability is negligible.

2. $G_{\mathrm{abo}}(s,b^*,x)=(c_1^{b^*}c_2)^x=\mathrm{Enc}_{pk}(0)$. Therefore, the image size of $G_{\mathrm{abo}}(s,b^*,\cdot)$ is at most $|R|$, and the amount of lossiness is at least $(\log|M|,\log|M|-\log|R|)$.

3. Hidden lossy branch: Since the scheme $\Pi=(\mathrm{Gen},\mathrm{Enc},\mathrm{Dec})$ is CPA-secure, the scheme $\Pi'=(\mathrm{Gen}',\mathrm{Enc}',\mathrm{Dec}')$, whose encryption algorithm is

$$\mathrm{Enc}_{pk}'(x_1,x_2)=(\mathrm{Enc}_{pk}(x_1),\mathrm{Enc}_{pk}(x_2))$$

and decryption algorithm is corresponding to it, is also CPA-secure. Now if $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ is a PPT distinguisher who can distinguish a lossy branch with non-negligible probability when given the function index s. We can use

$\mathcal{A}$ to construct a PPT adversary $\mathcal{A}'$ who breaks the CPA-secure of Π'. $\mathcal{A}'$ works as follows:

$\mathcal{A}'$:
on input pk;
obtains (b_0^*, b_1^*) from $\mathcal{A}$;
samples randomly $x_1^0, x_2^0, x_1^1, x_2^1 \in M$ such that $b_0^* x_1^0 + x_2^0 = 0$, $b_1^* x_1^1 + x_2^1 = 0$;
outputs the message $m_0 = (x_1^0, x_2^0), m_1 = (x_1^1, x_2^1)$ to its challenger;
obtains the challenge ciphertext $c^* = (c_1^b, c_2^b)$;
gives (pk, c_1^b, c_2^b) to $\mathcal{A}$;
outputs the bit b' that $\mathcal{A}$ outputs.

Therefore, we have

$$\Pr[\mathcal{A} \text{ success}] = \Pr[\mathcal{A}' \text{ success}].$$

This contradicts the CPA-security of Π'. ■

V. A Simple Implementation

In this section, we give a simple homomorphic encryption scheme which can be used to implement our constructions of almost-always chameleon ABO-TDFs and ABO-TDFs.

Formally, we consider the Damgård-Jurik (DJ) homomorphic encryption scheme [22]. Let $\mathrm{GenModulus}$ be a polynomial-time algorithm that, on input 1^λ, outputs (N, P, Q) where $N = PQ$ and P, Q are λ-bit primes (except with probability negligible). In addition, we require that $\gcd(N, \phi(N)) = 1$ (such N is called *admissible*). Now we describe the Damgård-Jurik scheme $\Pi = (\text{DJ.Gen}, \text{DJ.Enc}, \text{DJ.Dec})$:

- DJ.Gen(1^λ): runs $\mathrm{GenModulus}(1^\lambda)$ to obtain (N, P, Q). Then choose a natural number $\ell < P, Q$. The public key is (N, ℓ) and the private key is $sk = \mathrm{lcm}(P-1, Q-1)$.
- DJ.Enc : To encrypt a message $m \in \mathbb{Z}_{N^\ell}$ with respect to the public key (N, ℓ), choose randomly $r \leftarrow \mathbb{Z}_N^*$ and output the ciphertext
$$c = (1+N)^m \cdot r^{N^\ell} \bmod N^{\ell+1}.$$
- DJ.Dec : To decrypt a ciphertext c using the private key $sk = \mathrm{lcm}(P-1, Q-1)$, we first compute, by the Chinese Remainder Theorem d, such that $d = 1$ mod N^ℓ and $d = 0$ mod sk. Then compute $c^d = (1+N)^m$ mod $N^{\ell+1}$. At last, we can compute (m mod N^ℓ) using the algorithm given in [22].

In addition, [22] also proved that based on decisional composite residuosity assumption, the encryption scheme described above is CPA-secure.

We remark that the DJ-scheme is sufficient for us to construct the primitives introduced in this paper. In order to illustrate the main idea, we only give the concrete method to construct the ABO-TDFs of Section IV-B. Formally, a collection of ABO-TDFs $(S_{\text{abo}}, G_{\text{abo}}, G_{\text{abo}}^{-1})$ based on DJ-scheme is defined as follows:

- $S_{\text{abo}}(1^\lambda, b^*)$: It invokes DJ.Gen$(1^\lambda)$ to generate the public/private key pair (pk, sk) =$((N, \ell)$,(lcm$(P-1, Q-1)))$. Then it samples randomly $x_1, x_2 \in \mathbb{Z}_{N^\ell}$ satisfying $b^* x_1 + x_2 = 0$ and computes
$$c_1 = \text{DJ.Enc}_{pk}(x_1), c_2 = \text{DJ.Enc}_{pk}(x_2).$$
The function index is $s = (pk, c_1, c_2)$ and the trapdoor is $t = (sk, x_1, x_2)$.
- Evaluating a function G_{abo}: takes as input (s, b, x), where $(s, t) \leftarrow$ DJ.Gen(1^λ), $b, x \in \mathbb{Z}_{N^\ell}$, it computes
$$y = G_{\text{abo}}(s, b, x) = (c_1^b c_2)^x \bmod N^\ell$$
and outputs y.
- Inverting an injective function G_{abo}^{-1} : takes as input (t, b, y), where $b \neq b^*$, it computes and outputs
$$x = \text{DJ.Dec}_{sk}(y)(bx_1 + x_2)^{-1} \bmod N^\ell.$$
Note that, with overwhelming probability, the item $bx_1 + x_2$ mod N^ℓ has multiplicative inverse.

VI. Conclusions

In this paper, we generalize the construction of chameleon ABO-TDFs in [9]. As an application of our generalization, we construct an ABO-TDFs which can be used to construct CCA-secure public key encryption schemes by using homomorphic encryption scheme with some additional properties.

Acknowledgment

The authors would like to thank the anonymous referee(s) for suggestions of improving the writing of the paper.

References

[1] Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008. pp. 187-196. ACM, New York (2008)
[2] Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. Full version of [1]. `http://people.csail.mit.edu/cpeikert/pubs/lossy_tdf.pdf` or `http://eprint.iacr.org/2007/279.pdf`.
[3] Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. SIAM J. Comput. Vol. 40(6), pp. 1803-1844. (2011)
[4] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: PKC 2010. LNCS, vol. 6056, pp. 279-295. Springer, Heidelberg (2010)
[5] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. Journal of Cryptology. 26(1), pp. 39-74. (2013)
[6] Hemenway, B., Ostrovsky, R.: Lossy Trapdoor Functions From Smooth Homomorphic Hash Proof Systems. In: ECCC 2009. LNCS, vol. 16(127) (2009)
[7] Hemenway, B., Ostrovsky, R.: Building Lossy Trapdoor Functions from Lossy Encryption. In: ASIACRYPT 2013. LNCS, vol. 8270, pp. 241-260. Springer, Heidelberg (2013)
[8] Rosen, A., Segev, G.: Chosen-Ciphertext Security Via Correlated Products. In: TCC 2009. LNCS, vol. 5444, pp. 419-436. Springer, Heidelberg (2009)

[9] Lai, J., Deng R. and Liu S.: Chameleon All-But-One TDFs and Their Application to Chosen Ciphertext Security. In: PKC 2011. pp. 228-245. Springer, Heidelberg (2011)
[10] Liu Shengli, Lai Junzuo, and Deng Robert H.: General Construction of Chameleon All-But-One Trapdoor Functions. Journal of Internet Services and Information Security (JISIS), 1(2/3), 74-88 (2011)
[11] Hofheinz, D.: All-But-Many Lossy Trapdoor Functions. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 209-227. Springer, Heidelberg (2012)
[12] Xue, H., Li, B., Lu, X., Jia, D. and Liu, Y.: Efficient Lossy Trapdoor Functions Based on Subgroup Membership Assumptions. In: CANS 2013. LNCS, vol. 8257, pp. 235-250. Springer, Heidelberg (2013)
[13] Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: EUROCRYPT 2004. LNCS, vol. 3027, pp. 207-222. Springer, Heidelberg (2004)
[14] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: CRYPTO 1998. LNCS, vol. 1462, pp. 13-25. Springer, Heidelberg (1998)
[15] Cramer, R., Shoup, V.: Universal Hash Proofs and A Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 45-64. Springer, Heidelberg (2002)
[16] Dolev, D., Dwork, C., Naor, M.: Nonmalleable Cryptography. SIAM J. Comput. 30(2), 391-437 (2000)
[17] Matsuda, T., Hanaoka, G.: Chosen Ciphertext Security via Point Obfuscation. In: TCC 2014. LNCS, vol. 8349, pp. 95-120. Springer, Heidelberg (2014)
[18] Hu, C. and Liu, P.: An Enhanced Searchable Public Key Encryption Scheme with a Designated Tester and Its Extensions. Journal of Computers, Vol 7, No 3. pp. 716-723. (2012)
[19] Zhang, M., Wang, X., Li, W., and Yang, X.: CCA Secure Publicly Verfiable Public Key Encryption Without Pairings Nor Random Oracle and Its Applications. Journal of Computers. Vol. 8(8), pp. 1987-1994. (2013)
[20] Zhu, S. and Yang, X.: Proxy Re-encryption Scheme based on New Multivariate Quadratic Assumptions. Journal of Computers. Vol 8, No 12. pp. 3238-3242. (2013)
[21] Hemenway, B., Ostrovsky, R.: On Homomorphic Encryption and Chosen-Ciphertext Security. In: PKC 2012. pp. 52-65. Springer, Heidelberg (2012)
[22] Damgård, I., Jurik, M.: A Generalisation, A Simplification and Some Spplications of Pailliers Probabilistic Public-Key System. In: PKC 2001. LNCS, vol. 1992, pp. 119-136. Springer, Heidelberg (2001)

Jinyong Chang was born in Linfen, Shanxi province, China in 1982. He is a PH.D. student from Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interest is public key cryptography.

Rui Xue received the Ph.D degree in department of mathematics, Beijing Normal University, Beijing, China, 1999. Now he is a professor in State Key Lab of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. He is interest in the topics including provable security, public key cryptography, etc.