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

Jinyong Chang\textsuperscript{a,b}, Rui Xue\textsuperscript{a}

\textsuperscript{a} State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Email: \{changjinyong,xuerui\}@iie.ac.cn

\textsuperscript{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].
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 notations 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 \( m \sim E M \) denote picking element \( m \) uniformly at random from \( M \). Let \( \lambda \) be a security parameter and \( \text{PPT} \) 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 \( \text{neg}(\lambda) \) is negligible (in \( \lambda \)) if \( \text{neg}(\lambda) = o(\lambda^{-c}) \) for any constant \( c > 0 \). Let \( U_{\ell} \) denote the uniform distribution on \( \ell \)-bit binary strings.

B. Cryptographic Notions

1) Lossy Trapdoor Functions: Let \( n(\lambda) = \text{poly}(\lambda) \) be the length of the function’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 \((S_{\text{inj}}, S_{\text{loss}}, F, F^{-1})\) having the following properties.

1) Easy to sample an injective function with trapdoor: \( S_{\text{inj}}(1^\lambda) \) outputs \((s,t)\) where \( s \) is a function index and \( t \) is its trapdoor, \( F(s, \cdot) \) 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: \( S_{\text{loss}}(1^\lambda) \) 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 \( \{s : (s, t) \leftarrow S_{\text{inj}}(1^\lambda)\}_{\lambda \in \mathbb{N}} \) and \( \{s : \}

© 2014 ACADEMY PUBLISHER
3) Evaluation of lossy functions: For any \((a, b) \in A_\lambda \times B_\lambda\), if \((a, b) \in Q\) where \((s, t, Q) \leftarrow S_{ch}(1^\lambda)\), then \(F_{ch}(s, a, b, \cdot)\) 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 \(a \in A_\lambda\), \(\text{CLB}_{ch}\) computes a unique \(b \in B_\lambda\) to result in a lossy branch \((a, b)\). In formula, \(b \leftarrow \text{CLB}_{ch}(s, t, a)\) 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 \(A\) that receives \(s\) as input, where \((s, t, Q) \leftarrow S_{ch}(1^\lambda)\), 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 \(A\). Formally, let 
\[
A = (A_1, A_2)\]
be a distinguisher \(CH-I\) and define its advantage as
\[
\text{Adv}_{A}^{CH-I}(1^\lambda) = \Pr[\beta = \beta'] - \frac{1}{2}.
\]

6) Security (2)—Hard to find one-more lossy branch: Any probabilistic polynomial-time algorithm \(A\) that receives \((s, a, b)\) as input, where \((s, t, Q) \leftarrow S_{ch}(1^\lambda)\) and \((a, b) \notin Q\), has at most a negligible probability of outputting a pair \((a', b') \in Q \setminus \{(a, b)\}\).

If \(F_{ch}(t, a, b, \cdot)\) can correctly invert all images of \(g_{s,a,b}(\cdot)\) with \((a, b) \notin Q\) and \(\text{CLB}_{ch}(s, t, a)\) output \(b\) satisfying \((a, b) \in Q\), both with overwhelming probability, then we call the function \(ch\) the collection \(almost\-always\) chameleon ABO-TDFs.

4) Homomorphic Encryption:
Definition 6 (Homomorphic Encryption 91, 101):
A PKE scheme \(\Pi = (\text{Gen}, \text{Enc}, \text{Dec})\) is called homomorphic if:
- It is CPA secure;
- The plaintext space is a group \(M\) and we denote the group operation by \(\cdot^*\);
- 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
\[
\text{Enc}_{pk}(x_0 + x_1, r^*) = \text{Enc}_{pk}(x_0, r_0)\text{Enc}_{pk}(x_1, r_1).
\]

III. GENERIC CONSTRUCTION OF CHAMELEON ABO-TDFs

Let \(d = d(\lambda), k = k(\lambda), l = l(\lambda)\) be three polynomials of \(\lambda\). Let \(\Pi = (\text{Gen}, \text{Enc}, \text{Dec})\) 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 \(a, m \in M\), \((\text{Enc}_{pk}(m))^a = \text{Enc}_{pk}(am)\).

In the following, we give the description of our chameleon ABO-TDFs \((S_{ch}, F_{ch}, F_{ch}^{-1}, \text{CLB}_{ch})\) with the set of branches \(A \times B = \{A_\lambda \times B_\lambda\}_{\lambda \in \mathbb{N}} = \{M^k \times M^l\}_{\lambda \in \mathbb{N}}\):
- Sampling a function: \(S_{ch}\) takes as input \(1^\lambda\). It invokes \(\text{Gen}(1^\lambda)\) to generate \((pk, sk)\). Then it samples uniformly at random a matrix \(D = (A_{dx(k+1)}, B_{dx(k+1)}) = (x_{ij})_{d \times (k+1)}\) satisfying \(\text{rank}(B_{dx(k+1)}) \geq l\), where \(x_{ij} \in M\) for \(1 \leq i \leq d, 1 \leq j \leq k + l + 1\), and computes \(C = (c_{ij})_{d \times (k+1)} = (\text{Enc}_{pk}(x_{ij}))_{d \times (k+1)}\) 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 \((a_1, \ldots, a_k, b_1, \ldots, b_l) \in M^k \times M^l\) satisfying
\[
\begin{align*}
& a_{1}\ x_{11} + \ldots + a_{k}\ x_{1k} + x_{1,k+1} + b_1\ x_{1,k+2} + \ldots + b_l\ x_{1,k+l+1} = 0, \\
& a_{1}\ x_{21} + \ldots + a_{k}\ x_{2k} + x_{2,k+1} + b_1\ x_{2,k+2} + \ldots + b_l\ x_{2,k+l+1} = 0, \\
& \ldots, \\
& a_{1}\ x_{d1} + \ldots + a_{k}\ x_{dk} + x_{d,k+1} + b_1\ x_{d,k+2} + \ldots + b_l\ x_{d,k+l+1} = 0.
\end{align*}
\]
(1)

Evaluating a function: \(F_{ch}\) takes as input \((s, a_1, \ldots, a_k, b_1, \ldots, b_l, x)\), where \(x \in M\). It computes
\[
y = ((c_{11}^{(a_1)} \ldots c_{k1}^{(a_k)} c_{1,k+1}^{b_1} c_{1,k+2}^{b_2} \ldots c_{1,k+l+1}^{b_l}), \\
(c^{(a_1)}_{21} \ldots c^{(a_k)}_{k1} c^{(b_1)}_{2,k+1} c^{(b_2)}_{2,k+2} \ldots c^{(b_l)}_{2,k+l+1}), \\
\ldots, \\
(c^{(a_1)}_{d1} \ldots c^{(a_k)}_{dk} c^{(b_1)}_{d,k+1} c^{(b_2)}_{d,k+2} \ldots c^{(b_l)}_{d,k+l+1}))
\]
\[
:= (y_1, \ldots, y_d)
\]
and outputs \(y\).

Inverting an injective function: \(F_{ch}^{-1}\) takes as input \((s, t, a_1, \ldots, a_k, b_1, \ldots, b_l, y)\), where \((a_1, \ldots, a_k, b_1, \ldots, b_l) \notin Q\). Choose \(i \in \{1, \ldots, d\}\) satisfying
\[
a_{1}\ x_{i1} + \ldots + a_{k}\ x_{ik} + x_{i,k+1} + b_1\ x_{i,k+2} + \ldots + b_l\ x_{i,k+l+1} \neq 0
\]
and outputs
\[
x = \text{Dec}_{pk}(y_i) \cdot (a_{1}\ x_{i1} + \ldots + a_{k}\ x_{ik} + x_{i,k+1} + b_1\ x_{i,k+2} + \ldots + b_l\ x_{i,k+l+1})^{-1}
\]
Computing a lossy branch: \(\text{CLB}_{ch}\) takes as input \((s, t, a_1, \ldots, a_k)\). It can obtain \((b_1, \ldots, b_l)\) since \(\text{rank}(B_{dx(k+1)}) \geq l\), where \(B_{dx(k+1)}\) is the matrix that was sampled to satisfying (1). Then it output \((b_1, \ldots, b_l)\).

Now we state our main theorem as follows.

Theorem 7: The algorithms \((S_{ch}, F_{ch}, F_{ch}^{-1}, \text{CLB}_{ch})\) is a collection of \((\log |M|, \log |M| - \log |R|)\)-chameleon all-but-one trapdoor functions.
**Proof:** We prove this theorem from the following aspects:

1) About injective function $F_{ch}$: For any $(a_1, \ldots, a_k, b_1, \ldots, b_l) \in M^{k+l}$, if we know $(a_1, \ldots, a_k, b_1, \ldots, b_l) \not\in Q$, then

$$y = F_{ch}(s,a_1,\cdots,a_k,b_1,\cdots,b_l,x) = \left((c_1^1, \ldots, c_k^1, \cdots c_{k+1}^1, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x, \right.$$

$$\left. \cdots, (c_1^{a_1}, \ldots, c_k^{a_k}, \cdots c_d^{a_d}, c_{d+1}^{a_d}, c_{d+2}^{a_d}, \cdots c_{d+k+1}^{a_d}, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x \right)$$

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 $(a_1, \ldots, a_k, b_1, \ldots, b_l) \in M^{k+l}$, if $(a_1, \ldots, a_k, b_1, \ldots, b_l) \not\in Q$, then

$$y = F_{ch}(s,a_1,\cdots,a_k,b_1,\cdots,b_l,x) = \left((c_1^1, \ldots, c_k^1, \cdots c_{k+1}^1, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x, \right.$$

$$\left. \cdots, (c_1^{a_1}, \ldots, c_k^{a_k}, \cdots c_d^{a_d}, c_{d+1}^{a_d}, c_{d+2}^{a_d}, \cdots c_{d+k+1}^{a_d}, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x \right)$$

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 $c_{ij}$.

3) About lossy function $F_{ch}$: For any $(a_1, \ldots, a_k, b_1, \ldots, b_l) \in M^{k+l}$, if choose $(a_1, \ldots, a_k, b_1, \ldots, b_l) \not\in Q$, then

$$y = F_{ch}(s,a_1,\cdots,a_k,b_1,\cdots,b_l,x) = \left((c_1^1, \ldots, c_k^1, \cdots c_{k+1}^1, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x, \right.$$

$$\left. \cdots, (c_1^{a_1}, \ldots, c_k^{a_k}, \cdots c_d^{a_d}, c_{d+1}^{a_d}, c_{d+2}^{a_d}, \cdots c_{d+k+1}^{a_d}, b_l^1, b_l^2 \cdots b_l^{k+l+1})^x \right)$$

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 $c_{ij}$.

4) Hard to distinguish a lossy branch from an injective branch: Since the scheme $\Pi = (Gen, Enc, Dec)$ is CPA-secure, the scheme $\Pi' = (Gen', Enc', Dec')$, whose encryption algorithm is $Enc'_p(m_1, \ldots, m_d) = (Enc_p(m_1), \ldots, Enc_p(m_d))$ and decryption algorithm is corresponding to it, is also CPA-secure. Now if $A = (A_1, A_2)$ is a PPT distinguisher who can distinguish a lossy branch from an injective branch with non-negligible probability. We can use $A$ to construct a PPT adversary $A'$ who breaks the CPA-secure of $\Pi'$. $A'$ works as follows:

$A'$:

on input $pk$;

samples $x_1, \ldots, x_{d+k+1} \overset{R}{\leftarrow} M$ and $x_{d+k+1}' \overset{R}{\leftarrow} M$;

outputs $m_0 = (x_1, x_1, \ldots, x_{d+k+1})$;

receives the challenge ciphertext $c^* = (Enc'_p(m_0))^T$;

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

$$c_i = (Enc_p(x_{1i}), \ldots, Enc_p(x_{di}))^T,$$

for $i \in \{1, \ldots, k\}$, $k + 2, \ldots, k + l + 1$;

gives $s = (pk, C_b)$ to $A$ and obtains $(a_1, \ldots, a_k)$ from $A$;

computes $(b_1^0, \ldots, b_l^0)$ and $(b_1^1, \ldots, b_l^1)$ using $m_0, m_1$ and $(a_1, \ldots, a_k)$;

chooses randomly $\beta \in \{0, 1\}$ and gives $(b_1^\beta, \ldots, b_l^\beta)$ to $A$;

outputs the bit $\beta'$ which $A$ outputs.

Obviously, we have

$$Adv^{\text{CPA}}_{A'}(\lambda) \geq Adv^{\text{CHL}}_{A}(\lambda).$$
This contradicts the CPA-security of $\Pi'$.

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. $A$ could obtain the following equations:

$$
\begin{align*}
    a_1x_{11} + \cdots + a_kx_{1k} + x_{1,k+1} \\
    + b_1x_{1,k+2} + \cdots + b_x_{1,k+l+1} &= 0, \\
    a_1x_{21} + \cdots + a_kx_{2k} + x_{2,k+1} \\
    + b_1x_{2,k+2} + \cdots + b_x_{2,k+l+1} &= 0, \\
    \vdots \\
    a_1x_{d1} + \cdots + a_kx_{dk} + x_{d,k+1} \\
    + b_1x_{d,k+2} + \cdots + b_x_{d,k+l+1} &= 0.
\end{align*}
$$

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 a PPT algorithm $A$ that can find one-more lossy branch with non-negligible probability, then we can construct PPT inverter $I$ which can breaks the one-wayness of the PKE scheme. $I$ works as follows:

$I$:
- on input $y, pk$;
- chooses $x_{ij} \in M$ and computes $c_{ij} = \text{Enc}_pk(x_{ij})$ for $1 \leq i \leq d; 1 \leq j \leq k, k+2 \leq j \leq k+l+1$ and additionally chooses $(a_1, \ldots, a_k, b_1, \ldots, b_l) \leftarrow R M$;
- computes $c_{i,k+l+1} = a_i^0 \cdots a_i^k b_{k+2} \cdots b_{i,k+l+1}$, for $1 \leq i \leq d$;
- let $C$ be the matrix $(c_{ij})_{d \times (k+l+1)}$; chooses randomly $i_0 \in \{1, \ldots, d\}, j_0 \in \{1, \ldots, k+l+1\} \setminus \{k+1\}$; let $C_1$ be the matrix whose element $c_{i_0,j_0}$ is replaced by $y$ (without loss of generality we assume $j_0 \leq k$); gives $(s = (pk, C_1), (a_1, \ldots, a_k, b_1, \ldots, b_l))$ to $A$;
- if $A$ outputs $(a_1', \ldots, a_k', b_1', \ldots, b_l')$ and $a_{j_0} \neq a_{j_0}'$, then outputs

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

Obviously, we have

$$
\Pr[I \text{ success}] \geq \frac{1}{(k+l)d} - d \cdot \Pr[A \text{ success}]
$$

$$
= \frac{1}{(k+l)k_1} \Pr[A \text{ success}].
$$

This contradicts to the one-wayness of the PKE $\Pi$.

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 $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ 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 $a, m \in M$, $(\text{Enc}_pk(m))^a = \text{Enc}_pk(am)$.

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 $(S_{abo}, G_{abo}, G_{abo}^{-1})$ is defined as follows:

- $S_{abo}(1^k, b^r)$: It invokes $\text{Gen}(1^k)$ to generate $(pk, sk)$. Then it samples randomly $x_1, x_2 \in M$ satisfying $b^r_x x_1 + x_2 = 0$ and computes $c_1 = \text{Enc}_pk(x_1), c_2 = \text{Enc}_pk(x_2)$. 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 $(s, t) \leftarrow \text{Gen}(1^k)$, $b, x \in M$, it computes $y = G_{abo}(s, b, x) = (c_1^b c_2^x)$ and outputs $y$.

- Inverting an injective function $G_{abo}^{-1}$: takes as input $(t, b, y)$, where $b \neq b^r$, it computes

$$
x = \text{Dec}_pk(y)(bx + x_1)^{-1}.
$$

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 $b^r$, $G_{abo}(s, b, x) = (c_1^b c_2^x)$ computes a deterministic function over the domain $M$ since the randomness of $\text{Enc}_pk(\cdot)$ has been determined by the relationship between $x_1, x_2$ and $c_1, c_2$. Meanwhile, the function $G_{abo}(s, b, x)$ is always injective except that the event that $x_1 = 0$ whose probability is negligible.

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

3. Hidden lossy 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

$$
\text{Enc}_pk(x_1, x_2) = (\text{Enc}_pk(x_1), \text{Enc}_pk(x_2))
$$

and decryption algorithm is corresponding to it, is also CPA-secure. Now if $A = (A_1, A_2)$ is a PPT distinguisher who can distinguish a lossy branch with non-negligible probability when given the function index $s$. We can use...
A to construct a PPT adversary A’ who breaks the CPA-security of Π’. A’ works as follows:

\[ A’; \]
\[ \text{on input } pk; \]
\[ \text{obtains } (b_0^i, b_1^i) \text{ from } A; \]
\[ \text{samples randomly } x_0^i, x_2^i, x_1^i, x_1^j \in M \text{ such that } b_0^i x_0^i + x_2^i = 0, b_1^i x_1^i + x_2^i = 0; \]
\[ \text{outputs the message } m_0 = (x_1^i, x_2^i), m_1 = (x_1^j, x_2^i) \text{ to its challenger; } \]
\[ \text{obtains the challenge ciphertext } c^* = (c_1^*, c_2^*); \]
\[ \text{gives } (pk, c_1^*, c_2^*) \text{ to } A; \]
\[ \text{outputs the bit } b^* \text{ that } A \text{ outputs.} \]

Therefore, we have
\[ \Pr[A \text{ success}] = \Pr[A’ \text{ success}]. \]

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

- \( \text{DJ.Gen}(1^\lambda) \): runs GenModulus(1^\lambda) to obtain \( (N, P, Q) \). Then choose a natural number \( \ell < P, Q \). The public key is \((N, \ell)\) and the private key is \( sk = \text{lcm}(P - 1, Q - 1) \).

- \( \text{DJ.Enc} \) : To encrypt a message \( m \in \mathbb{Z}_N \) with respect to the public key \((N, \ell)\), choose randomly \( r \leftarrow \mathbb{Z}_N^* \) and output the ciphertext
\[ c = (1 + N)^m \cdot r^{N^{\ell+1}} \text{ mod } N^{\ell+1}. \]

- \( \text{DJ.Dec} \) : To decrypt a ciphertext \( c \) using the private key \( sk = \text{lcm}(P - 1, Q - 1) \), we first compute, by the Chinese Remainder Theorem \( d \), such that \( d - 1 \text{ mod } N^\ell \) and \( d = 0 \text{ mod } sk \). Then compute
\[ c^d = (1 + N)^m \text{ mod } N^{\ell+1}. \]

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 \((S_{abo}, G_{abo}, G_{abo})\) based on DJ-scheme is defined as follows:

- \( S_{abo}(1^\lambda, b^*) \): It invokes \( \text{DJ.Gen}(1^\lambda) \) to generate the public/private key pair \((pk, sk) = ((N, \ell), \text{lcm}(P - 1, Q - 1)))\). Then it samples randomly \( x_1, x_2 \in \mathbb{Z}_N \) 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 \((s, t)\), where \((s, t) \leftarrow \text{DJ.Gen}(1^\lambda)\), \( x \in \mathbb{Z}_N^* \), it computes
\[ y = G_{abo}(s, b, x) = (c_1^t c_2^x)^y \text{ mod } N^{\ell}. \]

- Inverting an injective function \( G_{abo}^{-1} \) : takes as input \((t, b, y)\), where \( b \neq b^* \), it computes and outputs
\[ x = \text{DJ.Dec}_{sk}(y)(bx_1 + x_2)^{-1} \text{ mod } N^{\ell}. \]

Note that, with overwhelming probability, the item \( bx_1 + x_2 \) mod \( N^{\ell} \) 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 CPA-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


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.