Chameleon All-But-One TDFs and Their Application to Chosen-Ciphertext Security

Junzuo Lai[1], Robert H. Deng[1], and Shengli Liu[2]

[1] School of Information Systems,
Singapore Management University, Singapore 178902
{junzuolai,robertdeng}@smu.edu.sg
[2] Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200240, China
slliu@sjtu.edu.cn

Abstract. In STOC'08, Peikert and Waters introduced a new powerful primitive called lossy trapdoor functions (LTDFs) and a richer abstraction called all-but-one trapdoor functions (ABO-TDFs). They also presented a black-box construction of CCA-secure PKE from an LTDF and an ABO-TDF. An important component of their construction is the use of a strongly unforgeable one-time signature scheme for CCA-security.

In this paper, we introduce the notion of chameleon ABO-TDFs, which is a special kind of ABO-TDFs. We give a generic as well as a concrete construction of chameleon ABO-TDFs. Based on an LTDF and a chameleon ABO-TDF, we presented a black-box construction, free of one-time signature, of variant of the CCA secure PKE proposed by Peikert and Waters.

Keywords: Chosen Ciphertext Security, Lossy Trapdoor Functions, Chameleon All-But-One Trapdoor Functions.

1 Introduction

Chosen-ciphertext security (CCA-security, for short) [33,14] is now considered as a standard notion of security for public key encryption (PKE) in practice. Numerous CCA-secure PKE schemes in the standard model, under both specific hardness assumption and general assumption, have been constructed over the years following several structural approaches.

The first approach for constructing CCA-secure PKE schemes was put forward by Naor and Yung [28]. As explained in [17], the approach employs a "two key" construction, where the well-formedness of a ciphertext is guaranteed by a (simulation-sound) non-interactive zero knowledge (NIZK) proof. The two-key/NIZK paradigm has led to CCA-secure PKE schemes based on general assumption [14], such as trapdoor permutations, and efficient schemes based on specific number theoretic assumptions [12,13], such as the decisional Diffie-Hellman (DDH) and composite residuosity assumptions.

Canetti, Halevi, and Katz [10] presented another approach for constructing CCA-secure PKE schemes using identity-based encryption (IBE) as a building

D. Catalano et al. (Eds.): PKC 2011, LNCS 6571, pp. 228–245, 2011.

block. The idea is to use, for each encryption, a fresh random verification key of a strongly unforgeable one-time signature scheme as the "identity" for IBE encryption. In order to tie the IBE ciphertext to this verification key, the ciphertext is signed using the corresponding signing key. Boneh and Katz [7] improved the efficiency of the scheme by using a MAC instead of a strongly unforgeable one-time signature. Some other efforts [8,25] further improved the efficiency.

The PKE schemes in [10,7,8,25] follow a similar method in the proof simulation. After the setup phase there is a certain set of well-formed ciphertexts that the simulator can decrypt corresponding to "identities" that the simulator knows the private keys. The remaining well-formed ciphertexts, that the simulator cannot decrypt corresponding to "identities" for which the simulator does not know the private keys, can be used as challenge ciphertexts in the simulation.

Recently, Peikert and Waters [31] introduced a new primitive called lossy trapdoor functions (LTDFs) and a richer abstraction called all-but-one trapdoor functions (ABO-TDFs). Peikert and Waters [31] constructed an elegant CCA-secure PKE scheme in a black-box manner based on an LTDF and an ABO-TDF. The scheme can be viewed as an application of the two-key paradigm [28], and the proof of security is similar to that of the IBE-based schemes [10]. An important component of their construction is the use of a strongly unforgeable one-time signature scheme for CCA-security, similar to that of Canetti, Halevi, and Katz [10]. This paper is motivated by improving the CCA-secure PKE construction of Peikert and Waters [31].

1.1 Our Contributions

Chameleon all-but-one TDFs. We introduce the notion of chameleon all-but-one TDFs (ABO-TDFs), which is a special kind of ABO-TDFs.

In an ABO-TDFs collection [31], each function has several *branches*. Almost all the branches are injective trapdoor functions, except for *one* branch which is lossy. Freeman et al. [18] generalized the definition of ABO trapdoor functions by allowing possibly *many* lossy branches (other than *one*).

As for chameleon ABO-TDFs, each function has many lossy branches just as the generalized definition in [18], but each branch is now represented by a pair (a, b). The "chameleon" property requires that for each a, it is easy to determine a unique b to come up with a lossy branch (a, b) with a trapdoor, while it is computationally indistinguishable to tell a lossy branch (a, b_0) from an injective branch (a, b_1) without the trapdoor.

Based on any CPA-secure PKE scheme with some additional property (mostly additively homomorphism), we propose a generic construction of chameleon ABO-TDFs. We can construct a chameleon ABO-TDF from any ABO-TDF in the sense of [31] and a chameleon hash function [24] targeting to the branch set. Yet the properties of the chameleon hash are a bit overkill for what we need and we build the needed properties directly into the constructions for better efficiency. In our construction, each chameleon ABO-TDF takes as input $((a, b), x)$, and outputs a ciphertext, which is an encryption of $x(ax_a + bx_b + x_d)$, where x_a, x_b, x_d are the trapdoor and the encryptions of x_a, x_b, x_d using the CPA-secure

PKE scheme are the public parameters of the function. Note that, due to the homomorphism of the CPA-secure PKE scheme, the chameleon ABO-TDF can be computed publicly. If (a, b) is a lossy branch, the chameleon ABO-TDF outputs an encryption of 0. Given a, with the trapdoor x_a, x_b, x_d, one can compute $b = (-ax_a - x_d) \cdot {x_b}^{-1}$, where (a, b) is a lossy branch. This computation requires that the message space of the PKE scheme is a finite field. In previous constructions of ABO-TDFs [31,18], each ABO-TDF takes as input (b, x), and outputs a ciphertext, which is an encryption of $x(b - b^*)$, where an encryption of b^* is the public parameter. The only lossy branch is $b = b^*$, and the ABO-TDF outputs an encryption of 0.

We also show how to instantiate the generic construction based on the Damgård-Jurik encryption scheme [15]. In fact, it is easy to transform the DDH-based all-but-one trapdoor function proposed by Freeman et al. [18] into a chameleon ABO-TDF using the same technique in our generic construction of chameleon ABO-TDFs.

CCA-secure PKE. We present a black-box construction of CCA-secure PKE based on an LTDF and a chameleon ABO-TDF. Our construction does not require strongly unforgeable one-time signature scheme, but a collision-resistant hash function, making it more efficient than that of Peikert and Waters [31].

The security proof of the construction does not rely on random oracles (RO) [5]. We follow a similar method of Peikert and Waters [31] in the proof simulation. In the security proof of Peikert and Waters's construction, when the adversary issues decryption queries, with overwhelming probability, the ABO-TDF works as an injective trapdoor function and the simulator uses the corresponding trapdoor to respond. In the challenge phase, the ABO-TDF works in lossy branch and the encrypted message is information hidden from the adversary. Because the ABO-TDF of Peikert and Waters has only *one* lossy branch, the simulator needs to know the lossy branch before the challenge phase and it resorts to strongly unforgeable one-time signature scheme.

In our CCA-secure PKE construction, the ABO-TDF is replaced by a chameleon ABO-TDF. Each chameleon ABO-TDF has many lossy branches (other than *one*) and each branch is represented by a pair (a, b). In our scheme, the first component of a branch a is correlated with the encrypted message, but b is independent of the message. Now, in the proof simulation, when the adversary submits the challenge messages, the simulator first computes a^*, which is correlated with the challenge messages, and computes b^* with the trapdoor of the chameleon ABO-TDF to make the branch (a^*, b^*) lossy. In other words, the simulator does not need to know the lossy branch before the challenge phase and it can generate a lossy branch after the adversary submits the challenge messages, which allows us to remove the requirement of strongly unforgeable one-time signature scheme. In our proof simulation, the simulator uses the same method of Peikert and Waters [31] to answer the adversary's decryption queries.

1.2 Related Work

Lossy trapdoor functions (LTDFs) were introduced by Peikert and Waters in [31]. Since their introduction, LTDFs have found many uses in cryptography. In particular, Peikert and Waters showed that any LTDF with enough lossiness can be used to construct an ABO-TDF, which can then be used to achieve CCA-security. In addition to CCA-secure encryption, LTDFs have been used in achieving deterministic encryption [3], lossy encryption [30], hedged public key encryption [2], security against selective opening attacks [4].

Peikert and Waters [31] presented constructions of LTDFs from the Decisional Diffie-Hellman (DDH) assumption and lattice assumptions. Boldyreva et al. [6] and Freeman et al. [18] gave (identical) efficient constructions of LTDFs from Paillier's decisional composite residuosity (DCR) assumption [29]. Freeman et al. [18] also gave efficient constructions of LTDFs based on composite residuosity assumption and d-Linear assumption [19,34]. Hemenway and Ostrovsky [20] showed that smooth homomorphic hash proof systems imply LTDFs. Kiltz et al. [23] showed that the RSA trapdoor function is lossy under the Φ-Hiding assumption of Cachin et al. [11]. Recently, Boyen and Waters [9] proposed two new discrete-log-type LTDFs, which are more efficient than earlier comparable constructions.

Rosen and Segev [32] showed that any collection of injective trapdoor functions that is secure under very natural correlated products can be used to construct a CCA-secure PKE scheme, and demonstrated that any collection of LTDFs with sufficient lossiness yields a collection of injective trapdoor functions that is secure under natural correlated products.

Mol and Yilek [27] extended the results of [31] and [32] and showed that only a non-negligible fraction of a single bit of lossiness is sufficient for building CCA-secure PKE schemes.

Hemenway and Ostrovsky [21] studied under which condition a homomorphic encryption implies CCA. They showed that a homomorphic encryption with cyclic plaintexts implies a family of LTDFs, and henceforth a CCA-secure encryption using the results of Peikert and Waters [31]. Our paper focuses on efficient construction of CCA-secure systems from families of LTDFs, compared with the construction of Peikert and Waters [31]. Our result is that we can do that with a special kind of LTDFs, namely, chameleon ABO-TDFs. A homomorphic encryption with cyclic plaintexts is not enough to construct a family of chameleon ABO-TDFs.

Recently, Kiltz et al. [22] introduced the notion of adaptive trapdoor functions (ATDFs) and a natural generalization they called tag-based adaptive trapdoor functions (TB-ATDFs). They showed that ATDFs and TB-ATDFs can be constructed directly using lossy+ABO-TDFs. They also showed that ATDFs and TB-ATDFs are strictly weaker than correlated-product trapdoor functions [32] and LTDFs [31]. They gave black-box constructions of CCA-secure PKE from both ATDFs and TB-ATDFs. The construction of CCA-secure PKE from TB-ATDFs is similar to the construction of Peikert and Waters [31]. But, compared with [31], one-time signature can be replaced by a MAC using the transform of

Boneh et al. [7]. They used the hardcore bit of the ATDF to construction one-bit PKE. But, if the given ATDF is a permutation or has linearly many simultaneous hardcore bits, they can use the ATDF as a key-encapsulation mechanism (KEM) for an CCA-secure symmetric encryption scheme to construction a CCA-secure PKE. In their construction of ATDF from lossy+ABO-TDF, the branch of ABO-TDF is the output of a target collision-resistant hash function, which takes the output of LTDF as input. The branch of ABO-TDF in their construction of TB-ATDF from lossy+ABO-TDF is a tag chosen randomly. As opposed to their construction, the first component of a branch (a, b) of chameleon ABO-TDF in our scheme a is chosen randomly, and the second component b is the output of a collision-resistant hash function, which takes the output of LTDF and $h(x) \oplus m$ as inputs, where m is the encrypted message. (Note that, our construction needs a fully collision resistant hash as opposed to target collision resistant as in [22].) This technique, which has already used in the RO model to construct CCA-secure PKE schemes [1], allows us to avoid using one-time signature, MAC, or other symmetric-key primitives.

Our works are also related to chameleon hash function [24]. Roughly speaking, chameleon hash functions are randomized collision-resistant hash functions with the additional property that given a trapdoor, one can efficiently generate collisions. Mohassel showed in [26] how to construct one-time signature from chameleon hash functions, which can be used in the construction of Peikert and Waters [31].

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we review some standard notations and cryptographic definitions. We introduce the notion of chameleon ABO-TDFs and present a generic construction and a concrete construction in Section 3. In Section 4, we present black-box constructions of CCA-secure PKE based on an LTDF and a chameleon ABO-TDF. Finally, we state our conclusion in Section 5.

2 Preliminaries

If S is a set, then $|S|$ denotes its size and $s \xleftarrow{\$} S$ denotes the operation of picking an element s uniformly at random from S . Let $\mathbb{N}$ denote the natural numbers. If $\lambda \in \mathbb{N}$ then 1^λ denotes the string of λ ones. Let $z \leftarrow \mathsf{A}(x, y, \ldots)$ denote the operation of running an algorithm A with inputs $(x, y, \ldots)$ and output z. Let U_ℓ denote the uniform distribution on ℓ-bit binary strings. A function $f(\lambda)$ is *negligible* if for every $c > 0$ there exists an λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

2.1 Hashing

Formally, a function $H : X \to Y$ is a collision-resistant (CR) hash function, if for all probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A}}(\lambda)$ is negligible in λ, where

$$\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A}}(\lambda) = \mathsf{Pr}[x, x' \leftarrow \mathcal{A}(H) : x' \neq x \wedge H(x') = H(x)].$$

A family of function $\mathcal{H} = \{h_i : X \to Y\}$ is pairwise independent, if for every distinct $x, x' \in X$ and every $y, y' \in Y$, $\mathsf{Pr}_{h \xleftarrow{\$} \mathcal{H}}[h(x) = y \text{ and } h(x') = y'] = 1/|Y|^2$.

2.2 Extracting Randomness

The *min-entropy* $\mathbf{H}_\infty(X)$ of a random variable X is $-\log(\max_x \mathsf{Pr}(X = x))$. Dodis, Reyzin and Smith [16] defined *average min-entropy* of X given Y to be the logarithm of the average probability of the most likely value of X given Y: $\tilde{\mathbf{H}}_\infty(X|Y) = -\log\left(\mathrm{E}_{y \leftarrow Y}\left[2^{-\mathbf{H}_\infty(X|Y=y)}\right]\right)$. They proved that if Y has 2^ℓ possible values and Z is any random variable, then $\tilde{\mathbf{H}}_\infty(X|(Y,Z)) \geq \mathbf{H}_\infty(X|Z) - \ell$.

The *statistical distance* between two probability distributions X and Y is $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_v |\mathsf{Pr}(X = v) - \mathsf{Pr}(Y = v)|$. Dodis, Reyzin and Smith [16] proved that if X, Y are random variables such that $X \in \{0,1\}^n$ and $\tilde{\mathbf{H}}_\infty(X|Y) \geq k$, and $\mathcal{H}$ is a family of pairwise independent hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$, then for $h \xleftarrow{\$} \mathcal{H}$, $\mathbf{SD}((Y, h, h(X)), (Y, h, U_\ell)) \leq \epsilon$ as long as $\ell \leq k - 2\log(1/\epsilon)$.

2.3 Public Key Encryption

A PKE scheme is a tuple of algorithms described as follows:

- $\mathsf{Kg}(\lambda)$ takes as input a security parameter λ. It outputs a public/private key pair $(\mathsf{PK}, \mathsf{SK})$.
- $\mathsf{Enc}(\mathsf{PK}, m)$ takes as input a public key PK and a message m. It outputs a ciphertext.
- $\mathsf{Dec}(\mathsf{SK}, c)$ takes as input a private key SK and a ciphertext c. It outputs a plaintext message or the special symbol $\perp$ meaning that the ciphertext is invalid.

We insist that all public key encryption schemes satisfy the obvious correctness condition (that decryption "undoes" encryption).

The strongest and commonly accepted notion of security for a PKE scheme is that of indistinguishability against an adaptive chosen ciphertext attack (CCA). It is defined using the following game between an adversary $\mathcal{A}$ and a challenger.

Setup. The challenger runs $\mathsf{Kg}(\lambda)$ to obtain a public/private key pair $(\mathsf{PK}, \mathsf{SK})$. It gives the public key PK to the adversary.

Query phase 1. The adversary $\mathcal{A}$ adaptively issues decryption queries c. The challenger responds with $\mathsf{Dec}(\mathsf{SK}, c)$.

Challenge. The adversary $\mathcal{A}$ submits two (equal length) messages m_0, m_1. The challenger selects a random bit $\beta \in \{0,1\}$, sets $c^* = \mathsf{Enc}(\mathsf{PK}, m_\beta)$ and sends c^* to the adversary as its challenge ciphertext.

Query phase 2. The adversary continues to adaptively issue decryption queries c, as in Query phase 1, but with the natural constraint that the adversary does not request the decryption of c^*.

Guess. The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0,1\}$ for β and wins the game if $\beta = \beta'$.

We define $\mathcal{A}$'s advantage in attacking the public key encryption scheme PKE with the security parameter λ as $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A}}(\lambda) = |\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$.

Definition 1. *A public key encryption scheme PKE is CCA secure, if for all polynomial-time adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A}}(\lambda)$ is negligible.*

The chosen-plaintext security CPA for a public key encryption scheme can be defined as the preceding game, except that adversary $\mathcal{A}$ is disallowed to issue any decryption query.

2.4 Lossy Trapdoor Functions

Informally, a collection of LTDFs is a collection of functions with two computationally indistinguishable branches: an injective branch with a trapdoor and a lossy branch losing information about its input.

Definition 2 (Lossy Trapdoor Functions). A collection of (n,k)-lossy trapdoor functions is a 3-tuple of (possibly probabilistic) polynomial-time algorithms $(\mathsf{G}, \mathsf{F}, \mathsf{F}^{-1})$ such that:

1. **Sampling an injective function:** $\mathsf{G}(1^\lambda, \mathsf{injective})$ outputs (s, td) where s is a function index and td is its trapdoor. The algorithm $\mathsf{F}(s,\cdot)$ computes a (deterministic) injective function $f_s(\cdot)$ over the domain $\{0,1\}^n$, and $\mathsf{F}^{-1}(s, td, \cdot)$ computes $f_s^{-1}(\cdot)$.
2. **Sampling a lossy function:** $\mathsf{G}(1^\lambda, \mathsf{lossy})$ outputs s where s is a function index. The algorithm $\mathsf{F}(s,\cdot)$ computes a (deterministic) function $f_s(\cdot)$ over the domain $\{0,1\}^n$ whose image has size at most 2^{n-k}.
3. **Hard to distinguish injective from lossy:** The ensembles $\{s : (s, td) \leftarrow \mathsf{G}(1^\lambda, \mathsf{injective})\}_{\lambda\in\mathbb{N}}$ and $\{s : s \leftarrow \mathsf{G}(1^\lambda, \mathsf{lossy})\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable.

3 Chameleon ABO-TDFs and Its Constructions

In this section, we first introduce our notion of chameleon ABO-TDFs. Then, based on a CPA-secure pubic key encryption scheme with some additional property, we propose a generic construction of chameleon ABO-TDFs. Finally, we instantiate the generic construction using the Damgård-Jurik encryption scheme [15].

3.1 Chameleon ABO-TDFs

The notion of ABO-TDFs, introduced by Peikert and Waters [31], is a richer abstraction of LTDFs. In an ABO collection, each function has several *branches*. Almost all the branches are injective trapdoor functions, except for *one* branch which is lossy. Freeman et al. [18] generalized the definition of ABO-TDFs by

allowing possibly *many* lossy branches (other than *one*). Let $\mathbb{B} = \{B_\lambda\}_{\lambda\in\mathbb{N}}$ be a collection of sets whose elements represent the branches, and we recall the definition of ABO-TDFs [18].

Definition 3(All-But-One Trapdoor Functions). A collection of (n, k)-all-but-one trapdoor functions is a 3-tuple of (possibly probabilistic) polynomial-time algorithms $(\mathsf{G}_{abo}, \mathsf{F}_{abo}, \mathsf{F}^{-1}_{abo})$ such that:

1. **Sampling a function:** For any $\lambda \in \mathbb{N}$ and $b^* \in B_\lambda$, $\mathsf{G}_{abo}(1^\lambda, b^*)$ outputs $(s, td, \tilde{S})$ where s is a function index, td is its trapdoor and $\tilde{S}$ is a set of lossy branches with $b^* \in \tilde{S} \subset B_\lambda$.
2. **Evaluation of injective functions:** For any $b \in B_\lambda$, if $b \notin \tilde{S}$ where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{abo}(1^\lambda, b^*)$, then $\mathsf{F}_{abo}(s, b, \cdot)$ computes a (deterministic) injective function $f_{s,b}(\cdot)$ over the domain $\{0,1\}^n$, and $\mathsf{F}^{-1}_{abo}(s, td, b, \cdot)$ computes $f^{-1}_{s,b}(\cdot)$.
3. **Evaluation of lossy functions:** For any $b \in B_\lambda$, if $b \in \tilde{S}$ where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{abo}(1^\lambda, b^*)$, then $\mathsf{F}_{abo}(s, b, \cdot)$ computes a (deterministic) function $f_{s,b}(\cdot)$ over the domain $\{0,1\}^n$ whose image has size at most 2^{n-k}.
4. **Security:** The ensembles $\{s : (s, td, \tilde{S}) \leftarrow \mathsf{G}_{abo}(1^\lambda, b^*_1)\}_{\lambda\in\mathbb{N}, b^*_1\in B_\lambda}$ and $\{s : (s, td, \tilde{S}) \leftarrow \mathsf{G}_{abo}(1^\lambda, b^*_2)\}_{\lambda\in\mathbb{N}, b^*_2\in B_\lambda}$ are computationally indistinguishable.
5. **Hard to find one-more lossy branch:** Any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives (s, b) as input, where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{abo}(1^\lambda, b^*)$ and $b \xleftarrow{\$} \tilde{S}$, has only a negligible probability of outputting an element $b' \in \tilde{S}\backslash\{b\}$.

We are now ready to introduce the notion of chameleon ABO-TDFs, which is a specific kind of ABO-TDFs with two variable (a, b) as a branch. The property we require is that given any a, it is easy to determine a unique lossy branch (a, b) with the help of a trapdoor, while (a, b_0) from a lossy branch family is computationally indistinguishable from (a, b_1) from an injective branch family with the trapdoor. We can construct a chameleon ABO-TDF from any ABO-TDF in the sense of [31] and a chameleon hash function [24] targeting to the branch set. Yet the properties of the chameleon hash are a bit overkill for what we need and we build the needed properties directly into the constructions for better efficiency.

Let $\mathbb{A} \times \mathbb{B} = \{A_\lambda \times B_\lambda\}_{\lambda\in\mathbb{N}}$ be a collection of sets whose elements represent the branches.

Definition 4 (Chameleon All-But-One Trapdoor Functions). A collection of (n, k)-chameleon all-but-one trapdoor functions is a 4-tuple of (possibly probabilistic) polynomial-time algorithms $(\mathsf{G}_{ch}, \mathsf{F}_{ch}, \mathsf{F}^{-1}_{ch}, \mathsf{CLB}_{ch})$ such that:

1. **Sampling a function:** For any $\lambda \in \mathbb{N}$, $\mathsf{G}_{ch}(1^\lambda)$ outputs $(s, td, \tilde{S})$ where s is a function index, td is its trapdoor and $\tilde{S} \subset A_\lambda \times B_\lambda$ is a set of lossy branches.

   Note that, a lossy branch is specified as a parameter to the function sampler of an ABO collection, but we have no such requirement.

2. **Evaluation of injective functions:** For any $(a,b) \in A_\lambda \times B_\lambda$, if $(a,b) \notin \tilde{S}$ where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{ch}(1^\lambda)$, then $\mathsf{F}_{ch}(s,a,b,\cdot)$ computes a (deterministic) injective function $g_{s,a,b}(\cdot)$ over the domain $\{0,1\}^n$, and $\mathsf{F}_{ch}^{-1}(s,td,a,b,\cdot)$ computes $g_{s,a,b}^{-1}(\cdot)$.
3. **Evaluation of lossy functions:** For any $(a,b) \in A_\lambda \times B_\lambda$, if $(a,b) \in \tilde{S}$ where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{ch}(1^\lambda)$, then $\mathsf{F}_{ch}(s,a,b,\cdot)$ computes a (deterministic) function $g_{s,a,b}(\cdot)$ over the domain $\{0,1\}^n$ whose image has size at most 2^{n-k}.
4. **Chameleon property:**
   (a) **Computing a lossy branch:** For any $a \in A_\lambda$, $\mathsf{CLB}_{ch}(s,td,a)$ computes a unique $b \in B_\lambda$ to result in a lossy branch (a,b). The uniqueness of b for a given a implies that any randomly chosen branch from $A_\lambda \times B_\lambda$ is injective with overwhelming probability.
   (b) **Hard to distinguish a lossy branch from an injective branch:** Any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives s as input, where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{ch}(1^\lambda)$, has only a negligible probability of distinguishing a pair $(a, b_0) \in \tilde{S}$ from $(a, b_1) \notin \tilde{S}$, even a is chosen by $\mathcal{A}$. Formally, Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a CH-LI distinguisher and define its advantage as

$$\mathsf{Adv}_{\mathcal{A}}^{\text{CH-LI}}(\lambda) = \left| \mathsf{Pr}\left[\beta = \beta' : \begin{array}{c} (s, td, \tilde{S}) \leftarrow \mathsf{G}_{ch}(1^\lambda); a \leftarrow \mathcal{A}_1(s); \\ b_0 = \mathsf{CLB}_{ch}(s, td, a); b_1 \xleftarrow{\$} B_\lambda; \\ \beta \xleftarrow{\$} \{0,1\}; \beta' \leftarrow \mathcal{A}_2(s, a, b_\beta) \end{array} \right] - \frac{1}{2} \right|.$$

   Given a collection of chameleon all-but-one trapdoor functions, it is hard to distinguish a lossy branch from an injective branch, if $\mathsf{Adv}_{\mathcal{A}}^{\text{CH-LI}}(\cdot)$ is negligible for every PPT distinguisher $\mathcal{A}$. This property implies that given a, without the trapdoor, the component b of the lossy branch (a,b) is distributed uniformly.
5. **Hard to find one-more lossy branch:** Any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives (s,a,b) as input, where $(s, td, \tilde{S}) \leftarrow \mathsf{G}_{ch}(1^\lambda)$ and $(a,b) \xleftarrow{\$} \tilde{S}$, has only a negligible probability of outputting a pair $(a', b') \in \tilde{S} \backslash \{(a,b)\}$. This property implies that the size of $\tilde{S}$ should not be too small.

In [31], Peikert and Waters also introduced a slightly relaxed definition of LTDFs, which they called *almost-always* LTDFs. Namely, there is only a negligible probability that $f_s(\cdot)$ is not injective or that $\mathsf{F}^{-1}(s,td,\cdot)$ incorrectly computes $f_s^{-1}(\cdot)$ for some input.

Similarly, we define *almost-always* chameleon ABO-TDFs. In a *almost-always* chameleon ABO-TDFs, *with overwhelming probability*, $\mathsf{F}_{ch}^{-1}(s,td,a,b,\cdot)$ inverts correctly on all values in the image of $g_{s,a,b}(\cdot)$ if $(a,b) \notin \tilde{S}$, and $\mathsf{CLB}_{ch}(s,td,a)$ outputs b such that $(a,b) \in \tilde{S}$.

3.2 Generic Construction

Let (Kg,Enc,Dec) be a CPA-secure PKE scheme, which is additively homomorphic. For the PKE scheme, we also assume that

1. $\mathcal{M}$ is its message space and $\mathcal{R}$ is its randomness space. Both spaces are large enough and $|\mathcal{M}| > |\mathcal{R}|$.
2. $\mathcal{M}$ is a finite field. For constructing *almost-always* chameleon ABO-TDFs, we only require that, $\mathcal{M}$ is a commutative ring with multiplicative identity and, *with overwhelming probability*, each element in $\mathcal{M}$ has multiplicative inverse (See the concrete construction in Section 3.3.).
3. $\mathsf{Enc}(\mathsf{PK}, m) \odot \mathsf{Enc}(\mathsf{PK}, m') = \mathsf{Enc}(\mathsf{PK}, m + m')$, where $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{Kg}(\lambda)$, m, $m' \in \mathcal{M}$, and $\odot$ denotes coordinate-wise multiplication of ciphertexts.
4. For $a, m \in \mathcal{M}$, $(\mathsf{Enc}(\mathsf{PK}, m))^a = \mathsf{Enc}(\mathsf{PK}, am)$, where exponentiation of a ciphertext is also coordinate-wise.

Now, we define a 4-tuple algorithms $(\mathsf{G}_{ch}, \mathsf{F}_{ch}, \mathsf{F}_{ch}^{-1}, \mathsf{CLB}_{ch})$ as follows:

1. **Sampling a function:** G_{ch} takes as input 1^λ, where λ is a security parameter. It first generates a keypair for the public key encryption scheme: $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{Kg}(\lambda)$. It then chooses $x_a, x_b, x_d \xleftarrow{\$} \mathcal{M}$ and computes
$$c_a = \mathsf{Enc}(\mathsf{PK}, x_a),\ c_b = \mathsf{Enc}(\mathsf{PK}, x_b),\ c_d = \mathsf{Enc}(\mathsf{PK}, x_d).$$
The function index is $s = (\mathsf{PK}, c_a, c_b, c_d)$, the trapdoor is $td = (\mathsf{SK}, x_a, x_b, x_d)$ and the set of lossy branches $\tilde{S}$ is all pairs $(a, b) \in \mathcal{M} \times \mathcal{M}$ such that $ax_a + bx_b + x_d = 0$.
2. **Evaluating a function:** F_{ch} takes as input (s, a, b, x), where $s = (\mathsf{PK}, c_a, c_b, c_d)$ is a function index and $x \in \mathcal{M}$. It computes $y = \left((c_a)^a \odot (c_b)^b \odot c_d\right)^x$, and outputs y.
3. **Inverting an injective function:** F_{ch}^{-1} takes as input (s, td, a, b, y), where $s = (\mathsf{PK}, c_a, c_b, c_d)$ is a function index, $td = (\mathsf{SK}, x_a, x_b, x_d)$ is the trapdoor and $(a, b) \notin \tilde{S}$. It computes $x = \mathsf{Dec}(\mathsf{SK}, y) \cdot (ax_a + bx_b + x_d)^{-1}$, and outputs x.
4. **Computing a lossy branch:** CLB_{ch} takes as input (s, td, a), where $s = (\mathsf{PK}, c_a, c_b, c_d)$ is a function index and $td = (\mathsf{SK}, x_a, x_b, x_d)$ is the trapdoor. It computes $b = (-ax_a - x_d) \cdot {x_b}^{-1}$, and outputs b.

Theorem 1. *The algorithms described above give a collection of* $(\log|\mathcal{M}|, \log|\mathcal{M}| - \log|\mathcal{R}|)$*-chameleon all-but-one trapdoor functions.*

Proof. We observe that, if (a, b) is not a lossy branch, namely $b \neq \mathsf{CLB}_{ch}(s, td, a) = (-ax_a - x_d) \cdot {x_b}^{-1}$, then $\mathsf{F}_{ch}(s, a, b, x)$ computes

$$\begin{aligned} y &= \left((c_a)^a \odot (c_b)^b \odot c_d\right)^x = \left((\mathsf{Enc}(\mathsf{PK}, x_a))^a \odot (\mathsf{Enc}(\mathsf{PK}, x_b))^b \odot \mathsf{Enc}(\mathsf{PK}, x_d)\right)^x \\ &= \mathsf{Enc}(\mathsf{PK}, x(ax_a + bx_b + x_d)), \end{aligned}$$

and $\mathsf{F}_{ch}^{-1}(s, td, a, b, y)$ computes

$$\begin{aligned} \mathsf{Dec}(\mathsf{SK}, y) \cdot (ax_a + bx_b + x_d)^{-1} &= \mathsf{Dec}(\mathsf{SK}, \mathsf{Enc}(\mathsf{PK}, x(ax_a + bx_b + x_d))) \\ &\quad \cdot (ax_a + bx_b + x_d)^{-1} \\ &= x(ax_a + bx_b + x_d) \cdot (ax_a + bx_b + x_d)^{-1} = x. \end{aligned}$$

So, we have shown invertibility for injective functions via the trapdoor information. Next, we show that if (a, b) is a lossy branch, namely $b = \mathsf{CLB}_{ch}(s, td, a) = (-ax_a - x_d) \cdot x_b^{-1}$, then F_{ch} evaluates a lossy function. In this case, F_{ch} computes

$$\begin{aligned}\left((c_a)^a \odot (c_b)^b \odot c_d\right)^x &= \left((\mathsf{Enc}(\mathsf{PK}, x_a))^a \odot (\mathsf{Enc}(\mathsf{PK}, x_b))^b \odot \mathsf{Enc}(\mathsf{PK}, x_d)\right)^x \\ &= \mathsf{Enc}(\mathsf{PK}, 0),\end{aligned}$$

and most of the information on the input is lost. The function $\mathsf{F}_{ch}(s, a, b, \cdot)$ is defined over the domain $\mathcal{M}$, and if $(a, b) \in \tilde{S}$, F_{ch} is a lossy function and the image size is at most $|\mathcal{R}|$. Therefore the amount of lossiness is at least $\log |\mathcal{M}| - \log |\mathcal{R}|$.

Given the public key encryption scheme (Kg,Enc,Dec) is CPA secure, it is easy to see that any probabilistic polynomial-time algorithm $\mathcal{A}$ has only a negligible probability of distinguishing a pair $(a, b_0) \in \tilde{S}$ from $(a, b_1) \notin \tilde{S}$, even a is chosen by $\mathcal{A}$.

Finally, we show that any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives (s, a, b) as input, where $(a, b) \in \tilde{S}$, has only a negligible probability of outputting a pair $(a', b') \in \tilde{S} \backslash \{(a, b)\}$. To see this, observe that the values x_a, x_b and x_d are initially hidden by the CPA secure public key encryption scheme. $\mathcal{A}$ could obtain the information that $ax_a + bx_b + x_d = 0$. However, there are exactly $|\mathcal{M}|^2$ pairs that satisfy this equation and each of them are equally likely. Thus, the adversary can output a pair (a', b') satisfying $(a', b') \neq (a, b)$ and $a'x_a + b'x_b + x_d = 0$ with negligible probability.

The formal proofs of the hardness of distinguishing a lossy branch from an injective branch of the chameleon ABO-TDFs, which can be reduced to the CPA security of the PKE scheme, and the hardness of finding one-more lossy branch, which can be reduced to the one-wayness of the PKE scheme, will be given in the full version of the paper.

3.3 A Concrete Construction

Based on the Damgård-Jurik encryption scheme [15], which is additively homomorphic, we present a concrete construction of *almost-always* chameleon ABO-TDFs by instantiating the generic construction. We begin with a brief description of the Damgård-Jurik encryption scheme [15], and then describe our construction.

Consider a modulus $N = PQ$, where P and Q are odd primes and $\gcd(N, \phi(N)) = 1$. Such an N is called *admissible* by Damgård and Jurik [15]. The following theorem was proved in [15]:

Theorem 2. *For any admissible N and a natural number $\ell < P, Q$, the map $\psi_\ell : \mathbb{Z}_{N^\ell} \times \mathbb{Z}_N^* \to \mathbb{Z}_{N^{\ell+1}}^*$ defined by $\psi_\ell(x, r) = (1 + N)^x r^{N^\ell} \mod N^{\ell+1}$ is an isomorphism, where*

$$\psi_\ell(x_1 + x_2 \mod N^\ell, r_1 r_2 \mod N) = \psi_\ell(x_1, r_1) \cdot \psi_\ell(x_2, r_2) \mod N^{\ell+1}.$$

Moreover, it can be inverted in polynomial time given $lcm(P - 1, Q - 1)$.

The following describes the Damgård-Jurik encryption scheme [15].

$\mathsf{DJ.Kg}(\lambda)$ Given the security parameter λ, choose an admissible modulus $N = PQ$ and a natural number $\ell < P, Q$. The published public key is $\mathsf{PK} = (N, \ell)$, and the private key is $\mathsf{SK} = \mathrm{lcm}(P-1, Q-1)$.

$\mathsf{DJ.Enc}(\mathsf{PK}, m)$ Given PK and a message $m \in \mathbb{Z}_{N^\ell}$, choose $r \xleftarrow{\$} \mathbb{Z}_N^*$ and output $c = (1+N)^m r^{N^\ell} \mod N^{\ell+1}$.

$\mathsf{DJ.Dec}(\mathsf{SK}, c)$ Given $\mathsf{SK} = \mathrm{lcm}(P-1, Q-1)$ and a ciphertext c, apply the inversion algorithm provided by Theorem 2 to compute $(m, r) = \psi_\ell^{-1}(\mathsf{SK}, c)$, and output m.

Damgård and Jurik [15] also proved that based on decisional composite residuosity assumption, the encryption scheme described above is CPA secure.

Now, given a Damgård and Jurik encryption scheme with algorithms $\mathsf{DJ.Kg}$, $\mathsf{DJ.Enc}$ and $\mathsf{DJ.Dec}$, we define a 4-tuple algorithms $(\mathsf{G}_{ch}, \mathsf{F}_{ch}, \mathsf{F}_{ch}^{-1}, \mathsf{CLB}_{ch})$ as follows:

1. **Sampling a function:** G_{ch} takes as input 1^λ, where λ is a security parameter. It runs $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{DJ.Kg}(\lambda)$, where $\mathsf{PK} = (N, \ell)$, and chooses $x_a, x_b, x_d \xleftarrow{\$} \mathbb{Z}_{N^\ell}$. Next, it computes

$$c_a = \mathsf{DJ.Enc}(\mathsf{PK}, x_a),\ c_b = \mathsf{DJ.Enc}(\mathsf{PK}, x_b),\ c_d = \mathsf{DJ.Enc}(\mathsf{PK}, x_d).$$

   The function index is $s = (\mathsf{PK}, c_a, c_b, c_d)$, the trapdoor is $td = (\mathsf{SK}, x_a, x_b, x_d)$ and the set of lossy branches $\tilde{S}$ is all pairs $(a, b) \in \mathbb{Z}_{N^\ell} \times \mathbb{Z}_{N^\ell}$ such that $ax_a + bx_b + x_d = 0 \mod N^\ell$.
2. **Evaluating a function:** F_{ch} takes as input (s, a, b, x), where $s = (N, \ell, c_a, c_b, c_d)$, $a, b, x \in \mathbb{Z}_{N^\ell}$. It computes $y = \left((c_a)^a \cdot (c_b)^b \cdot c_d\right)^x \mod N^{\ell+1}$, and outputs y.
3. **Inverting an injective function:** F_{ch}^{-1} takes as input (s, td, a, b, y), where $s = (N, \ell, c_a, c_b, c_d)$, $td = (\mathsf{SK}, x_a, x_b, x_d)$ and $(a, b) \notin \tilde{S}$. It computes

$$x' = \mathsf{DJ.Dec}(\mathsf{SK}, y),$$

   and outputs $x = x' \cdot (ax_a + bx_b + x_d)^{-1} \mod N^\ell$.

   Note that, *with overwhelming probability*, $(ax_a + bx_b + x_d) \mod N^\ell$ has multiplicative inverse.
4. **Computing a lossy branch:** CLB_{ch} takes as input (s, td, a), where $s = (N, \ell, c_a, c_b, c_d)$, $td = (\mathsf{SK}, x_a, x_b, x_d)$ and $a \in \mathbb{Z}_{N^\ell}$. It computes

$$b = (-ax_a - x_d) \cdot {x_b}^{-1} \mod N^\ell,$$

   and outputs b.

Theorem 3. *Under the composite residuosity assumption, the algorithms described above give a collection of* $((n-1)\ell, (n-1)\ell - n)$*-almost-always chameleon all-but-one trapdoor functions.*

Proof. The CPA security of the Damgård-Jurik encryption scheme and Theorem 1 guarantee that the algorithms described above give a collection of *almost-always* chameleon ABO-TDFs. Thus, it only remains to bound the amount of lossiness.

The function $\mathsf{F}_{ch}(s, a, b, \cdot)$ is defined over the domain $\{0,1\}^{(n-1)\ell}$, and if $(a, b) \in \tilde{S}$, F_{ch} is a lossy function and the image size is at most 2^n. Therefore the amount of lossiness is at least $(n-1)\ell - n$.

4 CCA-Secure PKE Scheme

Let $(\mathsf{G}, \mathsf{F}, \mathsf{F}^{-1})$ be a collection of (n, k_1)-lossy trapdoor functions, and let $(\mathsf{G}_{ch}, \mathsf{F}_{ch}, \mathsf{F}_{ch}^{-1}, \mathsf{CLB}_{ch})$ be a collection of (n, k_2)-chameleon all-but-one trapdoor functions having branches $\mathbb{A} \times \mathbb{B} = \{A_\lambda \times B_\lambda\}_{\lambda \in \mathbb{N}}$. Let $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$.

We assume that the public key encryption scheme has message space $\{0,1\}^\ell$. We also require that $k_1 + k_2 - n \geq k$ for some $k = \omega(\log n)$, and $\ell \leq k - 2\log(1/\epsilon)$ for some negligible ϵ (in λ).

Our PKE scheme consists of the following algorithms:

$\mathsf{Kg}(\lambda)$ Given the security parameter λ, generate an injective trapdoor function: $(s, td) \leftarrow \mathsf{G}(1^\lambda, \mathsf{injective})$. Then generate a chameleon all-but-one trapdoor function:$(s', td') \leftarrow \mathsf{G}_{ch}(1^\lambda)$. Finally, choose a collision-resistant hash function $H : \{0,1\}^* \rightarrow A_\lambda$ and $h \xleftarrow{\$} \mathcal{H}$. The published public key is $\mathsf{PK} = (s, s', H, h)$, and the private key is $\mathsf{SK} = (td, td')$.

$\mathsf{Enc}(\mathsf{PK}, m)$ Given PK and a message $m \in \{0,1\}^\ell$, choose $x \xleftarrow{\$} \{0,1\}^n, r \xleftarrow{\$} B_\lambda$ and compute $c_0 = h(x) \oplus m$, $c_1 = \mathsf{F}(s, x)$, $c_2 = \mathsf{F}_{ch}(s', t, r, x)$, where $t = H(c_0, c_1)$. Finally, output the ciphertext $c = (c_0, c_1, c_2, r)$.

$\mathsf{Dec}(\mathsf{SK}, c)$ Given $\mathsf{SK} = (td, td')$ and a ciphertext $c = (c_0, c_1, c_2, r)$, compute $x = \mathsf{F}^{-1}(s, td, c_1)$ and $t = H(c_0, c_1)$. Then check whether

$$c_1 = \mathsf{F}(s, x) \text{ and } c_2 = \mathsf{F}_{ch}(s', t, r, x).$$

If not, output $\perp$, else output $m = c_0 \oplus h(x)$.

It is clear that the above construction satisfies *correctness*. Our construction does not require strongly unforgeable one-time signature scheme. Compared with the scheme of Peikert and Waters [31], the ciphertext is compact without attached signature and decryption does not require performing signature verification. We now turn to security.

Theorem 4. *The algorithms* $(\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ *described above are a public key encryption scheme secure against adaptive chosen ciphertext attack.*

Proof. The proof is a sequence of games [35], $\mathrm{Game}_0, \mathrm{Game}_1, \ldots, \mathrm{Game}_5$, where Game_0 is the original adaptive chosen ciphertext attack game. Then we show that for all $i = 0, \ldots, 4$, Game_i and Game_{i+1} are (computationally) indistinguishable. Finally, we make an unconditional argument that an adversary must

have negligible advantage in Game_5. It follows that the public key encryption scheme is CCA-secure.

Game$_1$. We modify the way that the challenger computes the challenge ciphertext $c^* = (c_0^*, c_1^*, c_2^*, r^*)$ as

$$c_0^* = h(x^*) \oplus m_\beta,\ c_1^* = \mathsf{F}(s, x^*),\ c_2^* = \mathsf{F}_{ch}(s', t^*, r^*, x^*),$$

where $x^* \xleftarrow{\$} \{0,1\}^n$, $t^* = H(c_0^*, c_1^*)$ and $r^* = \mathsf{CLB}_{ch}(s', td', t^*)$.

Game$_2$. We modify the decryption oracle so that it reject all ciphertexts $c = (c_0, c_1, c_2, r)$, such that $r = r^*$ and $t = H(c_0, c_1) = t^*$.

Game$_3$. We modify the decryption oracle so that it applies the following *special rejection rule*: if the adversary submits a ciphertext $c = (c_0, c_1, c_2, r)$ for decryption, such that $r = \mathsf{CLB}_{ch}(s', td', t)$, where $t = H(c_0, c_1)$, then the decryption oracle immediately outputs reject and halts.

Game$_4$. This game is identical to **Game$_3$**, except for a small modification to the decryption oracle. When the adversary submits a ciphertext $c = (c_0, c_1, c_2, r)$ for decryption, the challenger computes $x = \mathsf{F}_{ch}^{-1}(s', td', c_2)$ and $t = H(c_0, c_1)$. Then it checks whether

$$c_1 = \mathsf{F}(s, x) \text{ and } c_2 = \mathsf{F}_{ch}(s', t, r, x).$$

If not, it outputs $\perp$, else outputs $m = c_0 \oplus h(x)$.

Game$_5$. In this game, we replace the injective function with a lossy one. Formally, in the **Setup** phase, we replace $(s, td) \leftarrow \mathsf{G}(1^\lambda, \text{injective})$ with $s \leftarrow \mathsf{G}(1^\lambda, \text{lossy})$.

Claim 1. *$Game_0$ and $Game_1$ are computationally indistinguishable, given the hardness of distinguishing a lossy branch from an injective branch of the chameleon all-but-one trapdoor functions collection.*

Proof. We prove this claim by describing a CH-LI distinguisher algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that receives s' as input where $(s', td') \leftarrow \mathsf{G}_{ch}(1^\lambda)$. The distinguisher $\mathcal{A}$ interacts with the adversary as follows.

In the **Setup** phase, $\mathcal{A}$ runs $(s, td) \leftarrow \mathsf{G}(1^\lambda, \text{injective})$, and chooses a collision-resistant hash function H and $h \xleftarrow{\$} \mathcal{H}$. The public key is $\mathsf{PK} = (s, s', H, h)$. We point out that $\mathcal{A}$ knows the injective trapdoor td, but does not know the trapdoor td' corresponding to s'.

When the adversary issues decryption queries, $\mathcal{A}$ responds using the injective trapdoor td. Note that, the only secret information that the decryption oracle needs to operate is td, which $\mathcal{A}$ knows.

When the adversary submits two (equal length) messages m_0, m_1, $\mathcal{A}$ flips a fair coin $\beta \in \{0,1\}$ and constructs the challenge ciphertext as follows:

1. It chooses $x^* \xleftarrow{\$} \{0,1\}^n$ and computes

$$c_0^* = h(x^*) \oplus m_\beta,\ c_1^* = \mathsf{F}(s, x^*),\ t^* = H(c_0^*, c_1^*).$$

2. Next, $\mathcal{A}_1$ submits t^* and sets the response as r^*. Then, it computes $c_2^* = \mathsf{F}_{ch}(s', t^*, r^*, x^*)$.
3. Finally, it outputs the ciphertext $c^* = (c_0^*, c_1^*, c_2^*, r^*)$.

When $r^* = \mathsf{CLB}_{ch}(s', td', t^*)$, $\mathcal{A}$ simulates Game_1 perfectly; otherwise, it simulates Game_0. Therefore, any difference in behavior between Game_0 and Game_1 immediately breaks the hardness of distinguishing a lossy branch from an injective branch of the chameleon all-but-one trapdoor functions collection.

Claim 2. *Game_1 and Game_2 are computationally indistinguishable, given the collision-resistant property of the hash function H.*

Proof. We observe that Game_1 and Game_2 behave equivalently unless an event E happens, which is that the adversary makes a legal (i.e., not equal to c^*) decryption query of the form $c = (c_0, c_1, c_2, r = r^*)$, where $t^* = H(c_0, c_1)$. We show that event E happens with negligible probability.

If event E happens, then because $c \neq c^*$ we must have $(c_0, c_1) \neq (c_0^*, c_1^*)$ and $H(c_0, c_1) = H(c_0^*, c_1^*) = t^*$. Therefore, we find a collision of the hash function H.

Because the hash function H is collision-resistant, we conclude that E happens with negligible probability.

Claim 3. *Game_2 and Game_3 are computationally indistinguishable, given the hardness of finding one-more lossy branch of the chameleon all-but-one trapdoor functions collection.*

Proof. We define the event F to be the event that the adversary makes a legal (i.e., not equal to c^*) decryption query of the form $c = (c_0, c_1, c_2, r)$, such that $r = \mathsf{CLB}_{ch}(s', td', t)$, where $t = H(c_0, c_1)$. It is clear that Game_2 and Game_3 proceed identically until event F occurs. We show that event F happens with negligible probability.

Note that, if $c \neq c^*$ and $(t, r) = (t^*, r^*)$, the decryption oracle rejects the ciphertext in both games. Therefore, if even F happens, then (t, r) is a new lossy branch of the chameleon all-but-one trapdoor function.

Because of the hardness of finding one-more lossy branch of the chameleon all-but-one trapdoor functions collection, we conclude that F happens with negligible probability.

Claim 4. *Game_3 and Game_4 are equivalent.*

Proof. The only difference between Game_3 and Game_4 is in the implementation of decryption oracle. We show that decryption oracle is equivalent in the two games.

In both games, when the adversary makes a legal (i.e., not equal to c^*) decryption query of the form $c = (c_0, c_1, c_2, r)$, where $t = H(c_0, c_1)$, the challenger checks that $c_1 = \mathsf{F}(s, x)$ and $c_2 = \mathsf{F}_{ch}(s', t, r, x)$ for some x that they compute (in different ways), and outputs $\perp$ if not.

Note that, if $r = \mathsf{CLB}_{ch}(s', td', t)$, the decryption oracle outputs rejects and halts in both games. Therefore, $\mathsf{F}(s, \cdot)$ and $\mathsf{F}_{ch}(s', t, r, \cdot)$ are both injective, and there is a unique x such that $(c_1, c_2) = (\mathsf{F}(s, x), \mathsf{F}_{ch}(s', t, r, x))$. Game_3 finds that x by computing $\mathsf{F}^{-1}(s, td, c_1)$, while Game_4 finds it by computing $\mathsf{F}_{ch}^{-1}(s', td', t, r, c_2)$.

Claim 5. *$Game_4$ and $Game_5$ are computationally indistinguishable, given the hardness of distinguishing injective functions from lossy functions of the lossy trapdoor functions collection.*

Proof. The only difference between Game_4 and Game_5 is in the **Setup** phase. In the **Setup** phase of Game_4, the challenger proceeds as in the original CCA game, outputting the public key $\mathsf{PK} = (s, s', H, h)$ where $(s, td) \leftarrow \mathsf{G}(1^\lambda, \mathsf{injective})$ and $(s', td') \leftarrow \mathsf{G}_{ch}(1^\lambda)$. In Game_5, **Setup** generates a lossy function instead, outputting $\mathsf{PK} = (s, s', H, h)$ where $s \leftarrow \mathsf{G}(1^\lambda, \mathsf{lossy})$ and $(s', td') \leftarrow \mathsf{G}_{ch}(1^\lambda)$.

We point out that the challenger knows the trapdoor td' of the chameleon all-but-one trapdoor function, but does not know the trapdoor td corresponding to s (if it even exists). The only secret information that the decryption oracle needs to operate is td', which the challenger knows.

It is straightforward to show that the adversary's views in the two games are indistinguishable, using the indistinguishability of injective and lossy functions.

Claim 6. *No (even unbounded) adversary has more than a negligible advantage in $Game_5$.*

We prove this claim by showing the fact that the value $h(x^*)$ is a nearly uniform and independent "one-time pad", and therefore the adversary has negligible advantage in guessing which message was encrypted.

Note that, in Game_5, both $\mathsf{F}(s, \cdot)$ and $\mathsf{F}_{ch}(s', t^*, r^*, \cdot)$ are lossy functions, and its image size is at most 2^{n-k_1} and 2^{n-k_2}, respectively. By the hypothesis that $k_1 + k_2 - n \geq k$, we have that the random variable $(c_1^*, c_2^*) = (\mathsf{F}(s, x^*), \mathsf{F}_{ch}(s', t^*, r^*, x^*))$ can take at most $2^{2n-k_1-k_2} \leq 2^{n-k}$ values.

Because x^* and h are independent, We also have $\tilde{\mathbf{H}}_\infty(x^*|(c_1^*, c_2^*, h)) \geq \mathbf{H}_\infty(x^*|h) - (n - k) = k$. Therefore, we have that $(c_1^*, c_2^*, h, h(x^*))$ $and(c_1^*, c_2^*, h, U_\ell)$ are within ϵ in statistical distance by our requirement that $\ell \leq k - 2\log(1/\epsilon)$, and we are done.

5 Conclusions

We introduced a new primitive called chameleon ABO-TDFs, which is a special kind of ABO-LTDFs. Given a CPA-secure public key encryption scheme with some additional property (mostly additively homomorphism), we also gave a generic and concrete construction of chameleon ABO-TDFs. Based on an LTDF and a chameleon ABO-TDF, we proposed a black-box construction of CCA-secure PKE which is more efficient than that of Peikert and Waters [31]. A future direction is to find other constructions and applications of chameleon ABO-TDFs.

Acknowledgement

We are grateful to the anonymous reviewers for their helpful comments. Our special thanks go to Adam O'Neill for helpful discussions and comments that improved the presentation of our paper. This work is partially funded by National Natural Science Foundation of China (No. 60873229) and Shanghai Rising-star Program (No. 09QA1403000), and also supported in part by A*Star SERC Grant No. 102 101 0027 in Singapore.

References

1. Abe, M., Kiltz, E., Okamoto, T.: Chosen Ciphertext Security with Optimal Ciphertext Overhead. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 355–371. Springer, Heidelberg (2008)
2. 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: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
3. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
4. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
5. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proc. of ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
6. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
7. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
8. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Proc. of ACM CCS 2005, pp. 320–329. ACM Press, New York (2005)
9. Boyen, X., Waters, B.: Shrinking the Keys of Discrete-Log-Type Lossy Trapdoor Functions. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 35–52. Springer, Heidelberg (2010)
10. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
11. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
12. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
13. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
14. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000); Preliminary version in STOC 1991
15. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001) (Full version with additional co-author J. B. Nielsen)
16. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)

17. Elkind, E., Sahai, A.: A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/042 (2002), http://eprint.iacr.org/
18. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)
19. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
20. Hemenway, B., Ostrovsky, R.: Lossy trapdoor functions from smooth homomorphic hash proof systems. In: ECCC, vol. 16(127) (2009)
21. Hemenway, B., Ostrovsky, R.: Homomorphic Encryption Over Cyclic Groups Implies Chosen-Ciphertext Security. Cryptology ePrint Archive, Report 2010/099 (2010)
22. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
23. Kiltz, E., O'Neill, A., Smith, A.: Lossiness of RSA and the chosen-plaintext security of OAEP without random oracles (2009) (manuscript)
24. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS 2000. The Internet Society, San Diego (2000)
25. Lai, J., Deng, R.H., Liu, S., Kou, W.: Efficient CCA-Secure PKE from Identity-Based Techniques. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 132–147. Springer, Heidelberg (2010)
26. Mohassel, P.: One-time Signatures and Chameleon Hash Functions. To appear in Proc. of SAC 2010. Springer, Heidelberg (2010)
27. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. Cryptology ePrint Archive, Report 2009/524 (2009)
28. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437. ACM, New York (1990)
29. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
30. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
31. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008, pp. 187–196. ACM, New York (2008)
32. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
33. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
34. Shacham, H.: A Cramer-Shoup encryption scheme from the Linear assumption and from progressively weaker Linear variants. Cryptology ePrint Archive, Report 2007/074 (2007)
35. Shoup, V.: Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive: Report 2004/332