Tweaking TBE/IBE to PKE Transforms with Chameleon Hash Functions

Rui Zhang

Research Center for Information Security (RCIS)
National Institute of Advanced Industrial Science and Technology (AIST)
r-zhang@aist.go.jp

Abstract. We present two transforms to acquire chosen ciphertext security from tag based techniques. The first one requires the separability of underlying primitives. By separability, informally, we mean the encryption algorithm has special structures and can process the identity and the message independently. Compared with generic transforms [8], it significantly reduces the ciphertext size overhead with only marginal computation cost. Compared with [11], the only known technique which directly achieves chosen ciphertext secure public key encryption from separable identity based primitives, it only requires selective-Tag/ID security of underlying primitives. Our second transform is less efficient but performs generically. Both transforms preserve the public verifiability of underlying primitives, and can be extended to hierarchical identity based encryption (HIBE) and threshold settings. As an independent interest, we also investigate the security requirements of chameleon hash functions to build strongly unforgeable one-time signatures.

1 Introduction

Indistinguishability against chosen ciphertext attack (CCA) is the standard security notion for public key encryption (PKE) schemes, which was established in [19,25,27,17,4]. While it is comparatively easy in the random oracle model [5], constructing a CCA-secure PKE is usually hard in the standard model. Up to now, there are only a few methods known to solve the problem in the standard model: Naor-Yung paradigm [25,17,28], universal hash proof [15,16,24], and an approach from tag/identity based encryption (TBE/IBE) techniques [9,13,10,22,8].

Along the last trend, most recently, Boyen, Mei and Waters [11] proposed two efficient techniques, referred as the BMW IBE transform and the BMW key encapsulation mechanism (KEM) transform, both of which achieve CCA-security from separable IBEs. Here, by separability, informally, we mean the encryption algorithm has special structures and can process the identity and the message independently. Another direct construction of KEM from a separable TBE was discovered by Kiltz [22].

Let's review the rough idea of the BMW PKE transform, which is based on separable IBEs. Write the encryption algorithm into two independent functions,

J. Katz and M. Yung (Eds.): ACNS 2007, LNCS 4521, pp. 323–339, 2007.

$f_1(params, m, r)$ and $f_2(params, id, r)$, where $params$ is the public system parameter, m is a plaintext, id is an arbitrary identity and r is the internal coin by the encryption algorithm. The public key of the PKE is then $params$ and the secret key is msk, which is the master secret key of the according IBE. For encryption, one computes $u = f_1(pk, m, r)$ then hashes u into an "identity" id by a collision resistant (or injective) hash function. The ciphertext is then $c = \langle u, v \rangle$, where $v = f_2(pk, id, r)$. Adaptive chosen-ID security is required here because the challenge identity is decided after the adversary selects the pair of chosen plaintexts. For decryption on $c = \langle u, v \rangle$, first reconstruct id from u, and decrypt c using the IBE decryption algorithm under identity id. Intuitively, to attack this PKE, an adversary may submit c' which maybe regarded as a ciphertext with different identity id' for the underlying IBE, however, this will not provide useful information to it, because of the semantic security of underlying IBEs.

Though the BMW PKE transform is very efficient in ciphertext size, the range of its application is quite limited, because it demands adaptive chosen-ID security. Additionally, note that the BMW PKE transform doesn't give a direct security reduction to that of the underlying IBE, since the simulator is supposed to submit the challenge identity with the pair of chosen plaintexts to its challenger, and the identity id is in fact determined by the challenge ciphertext. One may find it difficult to prove the security of the resulting PKE scheme sometimes, e.g., the BMW PKE transform + Gentry IBE [18].

We note that selective-Tag/ID security suffices for the TBE/IBE to KEM transform, because the challenge session key, thus the challenge identity, can be chosen even before interacting with the adversary. For the time being, we feel it no rush to build new KEMs, since BMW KEM [11] and Kiltz KEM [22] are already efficient enough for most applications requiring chosen ciphertext security at hand. On the other hand, we argue that such KEMs may be not publicly verifiable, thus may not fit into threshold settings in their original forms, which forms another motivation of this work.

Sketch of Our Ideas. Instead of a normal collision resistant hash function, we use a collision resistant chameleon hash function to construct the identity id. Interestingly, this simple modification achieves CCA-secure PKEs with public verifiability, whose security can be reduced to underlying IBEs. Recall a collision resistant chameleon hash function hashes a message x together with auxiliary randomness w, such that with a trapdoor, one can efficiently find a collision on (x', w'), such that (x, w) and (x', w') will be hashed to the same vale. While without the trapdoor, it is computationally infeasible to do so.

In the new strategy, for setup, the simulator generates a pair of public/secret keys (hk, td) for a chameleon hash function. The simulator then commits to an identity id^*, hashed from a dummy challenge ciphertext $\bar{u}$ with some randomness $\bar{r}$, and submits id^* to its challenger, who generates the public key and returns $params$ to the simulator. The public key of PKE now consists of $params$ and an additional hash key hk. For correctly reconstructing the identity id, the randomness r for hash function should be also appended to the ciphertext. Later upon receiving the real challenge ciphertext $c = \langle u^*, v^* \rangle$ from the IBE challenger, it

finds a collision (u^*, r^*) using td, i.e., both (u^*, r^*) and $(\bar{u}, \bar{r})$ will be hashed to id^*. In this way, the simulator can embed its own challenge into the challenge ciphertext for an IBE adversary, in other words, the security of the PKEs can be reduced to the underlying IBEs. Note that the BMW PKE transform doesn't give such a direct security reduction, since the target id is in fact determined by the challenge ciphertext itself. Alternatively, we can construct an adversary against the chameleon hash. Because this time the simulator does not have the trapdoor, while the challenge is partially determined according to the adversary, the chameleon hash needs to be collision resistant.

Moreover, as a supplement to our initial idea, a similar observation as [22] shows that a TBE scheme with selective-Tag and chosen ciphertext security suffices for the job. Since TBE is a possibly weaker primitive than IBE where the extract algorithm is not explicitly used, one may obtain efficient schemes using corresponding transforms. In fact, the BMW PKE scheme, an IBE private key is never generated, since it is more efficient to decrypt directly using the master key. Instantiate the method with Kiltz TBE [22], we obtain a corresponding PKE scheme with publicly verifiability and without pairings. Interestingly, all constructions of IBE schemes in the standard model in fact satisfy separability.

We go on to show how to remove the limitation of separability by sacrificing a little efficiency of above transform. Actually, a chameleon hash may also serve as a one-time signature by hashing with online key generation and switching. With similar idea of [13], one acquires a generic transform based on TBE/IBE and chameleon hash functions.

Compared with the BMW PKE transform, our transforms result in a slightly larger ciphertext overhead. We believe this is tolerable, since our requirement of underlying primitive is much weaker. Both transforms preserve the public verifiability of underlying primitives. Especially it transforms CPA-secure $(\ell+1)$-level HIBEs to CCA-secure ℓ-level HIBEs. It is worth noting that the chosen ciphertext security of resulting ℓ-level HIBE can be reduced to the semantic security of $(\ell+1)$-level HIBEs similarly.

Finally, as an independent interest, we investigate the necessary security requirements of building one-time strongly unforgeable signatures from chameleon hash functions. Although this intuition hides behind a lot of work, but it has not been formalized before.

Related Work. First CCA-secure construction of public key encryption, also known as Naor-Yung paradigm, was due to [25,17], and further generalized by Sahai [28], which is quite inefficient. Cramer and Shoup gave the first practical CCA-secure public key encryption scheme [15] in the standard model and later generalized to universal hash proof system [16]. These two approaches utilize certain non-interactive proofs of "well-formness". Apart from the above, Canetti, Halevi and Katz [13] presented a generic construction from weak IBEs, where no such non-interactive proofs are needed. Boneh and Katz [10] further improved the efficiency of [13]. The full version of [13,10] appeared as [8].

The notion of identity based encryption was due to Shamir, whose original intention was to simplify the management of public key certificates. Boneh

and Franklin [9] defined the first formal security notion for identity based encryption, namely indistinguishability against chosen-ID and chosen ciphertext attack (IND-ID-CCA) and gave the first functional scheme based on pairings. Independently, Cocks [14] proposed another identity based encryption based on decisional quadratic residue assumption. Canetti, Halevi and Katz [12] defined another useful security notion with weaker attack model, namely security against selective-ID attack and chosen plaintext/ciphertext attack (IND-sID-CPA/CCA). On the other hand, Gentry and Silverberg generated the notion of identity based encryption to hierarchical identity based encryption (HIBE).

Boneh and Boyen proposed two efficient IBE schemes (referred as BB1 and BB2) with selective-ID security [6]. They further generalized to adaptive chosen-ID security [7], but the scheme is quite inefficient. Waters subsequently presented the first practical IBE scheme [30]. Most recently, Gentry gave a more efficient IBE scheme, which however, relies on a very strong assumption [18].

Most recently, independently from our work, Abe, Cui, Imai and Kiltz [1] considered building tag-KEM [2] from special ID-Based KEMs (IBKEMs), called partitioned IBKEMs, which are essentially the same as separability. A tag-KEM is more flexible in building secure hybrid encryptions, which differs slightly from our goal of building PKEs, however. We remark that we are additionally interested in directly building PKEs, from black-box TBEs/IBEs using chameleon hash functions.

Organizations. The rest of the paper will be arranged as follows: we give some definitions in Section 2, then give our transforms and analyze their securities in Section 3 and Section 4. We explain applications of our results in Section 5. Finally in Section 6, we give detailed comparisons among some typical schemes.

2 Preliminary

In this section, we give some notations and definitions, then review some hard problems related to pairings.

Notations. If x is a string, let $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If S is a set then $s \leftarrow S$ denotes the operation of picking an element s of S uniformly at random. We write $z \leftarrow \mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $(x, y, \ldots)$ and an output z. Denote $x||y$ as the string concatenation of x and y. If $k \in \mathbb{N}$, a function $f(k)$ is negligible if $\exists\ k_0 \in \mathbb{N}, \forall\ k > k_0,\ f(k) < 1/k^c$, where $c > 0$ is a constant.

2.1 Public Key Encryption

A public key encryption scheme consists of three algorithms $\mathcal{PKE} = (\mathsf{PKEkg}, \mathsf{PKEenc}, \mathsf{PKEdec})$. The randomized key generation algorithm taking a security parameter k as the input, generates a public key pk and a corresponding secret key sk, which is denoted as $(pk, sk) \leftarrow \mathsf{PKEkg}(1^{\mathsf{k}})$. The randomized encryption algorithm taking pk and a plaintext m taken from the message space as

inputs, with internal coin flipping r, outputs a ciphertext c, which is denoted as $c \leftarrow \mathsf{PKEenc}(pk, m, r)$, in brief $c \leftarrow \mathsf{PKEenc}(pk, m)$. The deterministic decryption algorithm taking sk and a ciphertext c as input, outputs the corresponding m, or "$\perp$" (indicating invalid ciphertext), denoted as $m \leftarrow \mathsf{PKEdec}(sk, c)$. We require a PKE scheme should satisfy the standard correctness requirement, namely for all $(pk, sk) \leftarrow \mathsf{PKEkg}(1^k)$ and all m, $\mathsf{PKEdec}(sk, \mathsf{PKEenc}(pk, m)) = m$.

IND-CCA-Security. We say a public key encryption scheme is (ϵ, q, T)-IND-CCA secure, if the advantage of any adversary $\mathcal{A}$ with at most q queries to a decryption oracle $\mathcal{DO}$, is at most ϵ within time T in the following experiment.

$$\mathrm{Adv}^{\mathsf{ind\text{-}cca}}_{\mathcal{PKE},\mathcal{A}} \stackrel{\mathsf{def}}{=} \Pr[(pk, sk) \leftarrow \mathsf{PKEkg}(1^k); (m_0, m_1, s) \leftarrow \mathcal{A}^{\mathcal{DO}}(pk); \\ b \leftarrow \{0,1\}; c^* \leftarrow \mathsf{PKEenc}(pk, m_b); b' \leftarrow \mathcal{A}^{\mathcal{DO}}(c^*, s) : b' = b] - 1/2$$

where $\mathcal{DO}$ returns the corresponding decryption result on a query on ciphertext c, whereas $\mathcal{A}$ is forbidden to query c^* at $\mathcal{DO}$. We say a PKE is IND-CCA-secure, if for polynomially bounded q and T, ϵ is negligible.

2.2 Tag Based Encryption

Informally, a tag based encryption (TBE) is a public key encryption scheme where the encryption and the decryption operations take an additional "tag" which is public binary string with proper length.

A TBE scheme consists of three algorithms $\mathcal{TBE} = (\mathsf{TBEkg}, \mathsf{TBEenc}, \mathsf{TBEdec})$. The randomized key generation algorithm TBEkg, taking a security parameter k as the input, outputs a public key pk and a corresponding secret key sk, denoted as $(pk, sk) \leftarrow \mathsf{TBEkg}(1^k)$. The randomized encryption algorithm TBEenc taking a public key pk, a tag t and a plaintext m taken from the message space as inputs, with internal coin flipping r, outputs a ciphertext c, which is denoted as $c \leftarrow \mathsf{TBEenc}(pk, t, m, r)$, in brief $c \leftarrow \mathsf{TBEenc}(pk, t, m)$. The deterministic algorithm TBEdec taking a secret key sk, a tag t and a ciphertext c as inputs, outputs a plaintext m, or "$\perp$" indicating invalid ciphertext, which is denoted $m \leftarrow \mathsf{TBEdec}(sk, t, m)$. We require for all $(pk, sk) \leftarrow \mathsf{TBEkg}(1^k)$, all m and all t, $\mathsf{TBEdec}(sk, t, \mathsf{TBEenc}(pk, t, m)) = m$.

Separability. A TBE is said to be sparable if the encryption algorithm can be arranged in two parts, such that one part is uniquely determined by pk, m and the random coin r, in brief $u \leftarrow f_1(pk, m, r)$, and the other part is uniquely determined by the pk, t and r, in brief $v \leftarrow f_2(pk, t, r)$. The ciphertext is $c = \langle u, v \rangle$.

IND-sTag-CCA-Security. We consider a weak security called indistinguishability against selective-tag and chosen ciphertext attack IND-sTag-CCA. Namely, the tag to be used in the challenge is chosen before the key generation phase. Again,

the scheme is (ϵ, q, T)-IND-sTag-CCA-secure if any adversary $\mathcal{A}$ with at most q queries to a decryption oracle $\mathcal{DO}$, has advantage at most ϵ within time T in the following experiment.

$$\mathrm{Adv}^{\text{ind-stag-cca}}_{\mathcal{TBE},\mathcal{A}} \stackrel{\text{def}}{=} \Pr[(t^*, s_0) \leftarrow \mathcal{A}(1^k); (pk, sk) \leftarrow \mathsf{TBEkg}(1^k); (m_0, m_1, s_1) \leftarrow \mathcal{A}^{\mathcal{DO}}(pk, s_0); b \leftarrow \{0,1\}; c^* \leftarrow \mathsf{TBEenc}(pk, t^*, m_b); b' \leftarrow \mathcal{A}^{\mathcal{DO}}(c^*, s_1) : b' = b] - 1/2$$

where $\mathcal{DO}$ returns the corresponding decryption result on a query of a ciphertext c under tag t, whereas $\mathcal{A}$ is forbidden to query any ciphertext under tag t^* at $\mathcal{DO}$. We say a TBE is IND-sTag-CCA-secure, if for polynomially bounded q and T, ϵ is negligible.

2.3 Identity Based Encryption

An identity based encryption (IBE) can be regarded as a special tag based encryption equipped with an additional extraction algorithm, with inputs a master secret key and an tag, outputs a secret key that is capable to decrypt ciphertext corresponding to this tag.

An IBE scheme consists of four algorithms $\mathcal{IBE} = (\mathsf{IBEkg}, \mathsf{IBEext}, \mathsf{IBEenc}, \mathsf{IBEdec})$. The randomized key generation algorithm IBEkg, taking a security parameter k as the input, outputs a public parameter $params$ and a master secret key msk, denoted as $(params, msk) \leftarrow \mathsf{TBEkg}(1^k)$. The extract algorithm, possibly randomized, takes inputs of $params$, msk and an identity id, outputs a secret key sk_{id} for id, denoted as $sk_{id} \leftarrow \mathsf{IBEext}(params, msk, id)$, in brief $sk_{id} \leftarrow \mathsf{IBEext}(msk, id)$. The randomized encryption algorithm TBEenc takes $params$, an identity id and a plaintext m taken from the message space as inputs, with internal coin flipping r, outputs a ciphertext c, which is denoted as $c \leftarrow \mathsf{IBEenc}(params, id, m, r)$, in brief $c \leftarrow \mathsf{IBEenc}(params, id, m)$. The deterministic algorithm IBEdec takes a secret key sk_{id}, an identity id and a ciphertext c as inputs, outputs a plaintext m, or a special symbol "$\perp$", which is denoted $m \leftarrow \mathsf{IBEdec}(sk_{id}, id, c)$. We require for all $(params, msk) \leftarrow \mathsf{IBEkg}(1^k)$, $sk_{id} \leftarrow \mathsf{IBEext}(msk, id)$ and all m, we have $\mathsf{IBEdec}(sk_{id}, id, \mathsf{IBEenc}(params, id, m)) = m$.

Separability. An IBE is said to be sparable if the encryption algorithm can be arranged in two parts, such that one part is uniquely determined by $params$, m and the random coin r, in brief $u \leftarrow f_1(params, m, r)$, and the other part is uniquely determined by the $params$, id and r, in brief $v \leftarrow f_2(params, id, r)$. The ciphertext is $c = \langle u, v \rangle$.

IND-sID-CPA-Security. We consider security of indistinguishability against selective-ID and chosen plaintext attack (IND-sID-CPA). We say an identity based encryption is (ϵ, q, T)-IND-sID-CPA-secure if the advantage of any adversary $\mathcal{A}$

is at most ϵ, with access q times to an extraction oracle $\mathcal{EO}$ within time T in the following experiment.

$$\begin{aligned}\mathrm{Adv}^{\mathsf{ind\text{-}sid\text{-}cpa}}_{\mathcal{IBE},\mathcal{A}} \stackrel{\mathsf{def}}{=} \Pr[(id^*, s_0) \leftarrow \mathcal{A}(1^k); (params, msk) \leftarrow \mathsf{IBEkg}(1^k);\\ (m_0, m_1, s_1) \leftarrow \mathcal{A}^{\mathcal{EO}}(params, s_0); b \leftarrow \{0,1\};\\ c^* \leftarrow \mathsf{IBEenc}(params, id^*, m_b); b' \leftarrow \mathcal{A}^{\mathcal{EO}}(c^*, s_1) : b' = b] - 1/2\end{aligned}$$

where $\mathcal{EO}$ returns the corresponding secret key on a query on identity id, whereas $\mathcal{A}$ is forbidden to query id^* at $\mathcal{EO}$. We say an IBE is IND-sID-CPA-Secure, if for polynomially bounded q and T, ϵ is negligible.

2.4 Digital Signature

A signature scheme consists of three algorithms $\mathcal{S} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$. The randomized key generation algorithm G takes a security parameter k, and generates signing key $sigk$ and verification key vk. The possibly randomized signing algorithm S takes as inputs $sigk$ and $m \in \{0,1\}^*$, where m is a message, and outputs a signature σ. The deterministic verification algorithm V takes as inputs vk, m and σ, and outputs a symbol $\beta \in \{\mathtt{accept}, \mathtt{reject}\}$, denoted as $\beta \leftarrow \mathsf{V}(vk, m, \sigma)$. We require that for all $(sigk, vk) \leftarrow \mathsf{G}(1^k)$, all $m \in \{0,1\}^*$, $\mathsf{V}(vk, m, \mathsf{S}(sigk, m)) = \mathtt{accept}$.

(Strong) Unforgeability. We consider strong unforgeability against adaptive chosen message attack sUF-CMA [3]. Let $\mathcal{S} = (\mathsf{G}, \mathsf{S}, \mathsf{V})$ be a signature scheme. Let $\mathcal{A}$ and k be an adversary and a security parameter, respectively.

Denote $\{(\sigma_i, m_i)\}_{q_s}$ as the set contains all q_s pairs of queries and replies between $\mathcal{A}$ and $\mathcal{SO}$, where $\mathcal{SO}$ is a signing oracle which for a given message m, returns a corresponding signature σ. The success probability of $\mathcal{A}$ is defined as

$$\begin{aligned}\mathrm{Suc}^{\mathsf{suf\text{-}cma}}_{\mathcal{S},\mathcal{A}}(k) \stackrel{\mathsf{def}}{=} \Pr[(vk, sigk) \leftarrow \mathsf{Gen}(1^k); (\sigma^*, m^*) \leftarrow \mathcal{A}^{\mathcal{SO}}(vk)\\ : \mathsf{V}(vk, m^*, \sigma^*) = \mathtt{accept} \wedge (\sigma^*, m^*) \notin \{(\sigma_i, m_i)\}_{q_s}]\end{aligned}$$

We say $\mathcal{S}$ is (t, ϵ)-sUF-CMA secure if for any $\mathcal{A}$ in time bound t, $\mathcal{A}$'s success probability is at most ϵ. Especially, we say that $\mathcal{S}$ is sUF-CMA secure if ϵ is negligible.

2.5 Collision Resistant Chameleon Hash Function

A collision resistant chameleon hash function consists of three algorithms $\mathcal{CMH} = (\mathsf{CMkg}, \mathsf{CMhash}, \mathsf{CMswch})$. The randomized key generation algorithm CMkg taking a security parameter k as the input, outputs a hash key hk and a trapdoor td, denoted as $(hk, td) \leftarrow \mathsf{CMkg}(1^k)$. The randomized hashing algorithm takes inputs a public key hk, an auxiliary random coin w drawn from space $\mathcal{R}$ and a value $x \in \{0,1\}^*$, outputs a binary string y of fixed length l, denoted as $y \leftarrow \mathsf{CMhash}(hk, x, w)$. The switch algorithm CMswch takes inputs the trapdoor

td, a pair of messages x, x', the corresponding auxiliary random coin w, outputs a pair of (x', w') with $x' \neq x$, such that $\mathsf{CMhash}(hk, x, w) = \mathsf{CMhash}(hk, x', w')$, denoted as $r' \leftarrow \mathsf{CMswch}(td, x, w, x')$. Finally, for all $x, x' \leftarrow \{0,1\}^*$ and $w \in \mathcal{R}$, we require $w' \leftarrow \mathsf{CMswch}(td, x, w, x')$ is uniformly distributed in $\mathcal{R}$ and we call this property the uniformness of a chameleon hash function. We next give two flavors of security requirements for a chameleon hash, namely collision resistance (CR) and oracle collision resistance (OCR).

Collision Resistance. We say a chameleon hash function is $(\epsilon_{\mathcal{H}}, T_{\mathcal{H}})$-collision resistant (CR) if any adversary $\mathcal{A}$ without access to the trapdoor td, the success probability of finding collisions is at most $\epsilon_{\mathcal{H}}$ within time T in the following experiment.

$$\begin{aligned}\mathrm{Succ}^{\mathsf{cr}}_{\mathcal{CMH},\mathcal{A}} \stackrel{\mathsf{def}}{=} \Pr[&(hk, td) \leftarrow \mathsf{CMkg}(1^k); x \leftarrow \mathcal{A}(hk);\\ &w \leftarrow \mathcal{R}; y \leftarrow \mathsf{CMhash}(hk, x, w); (x', w') \leftarrow \mathcal{A}(hk, x, w)\\ &: (x', w') \neq (x, w) \wedge y = \mathsf{CMhash}(hk, x', w')]\end{aligned}$$

We say a chameleon hash function is collision resistant, if for polynomially bounded $T_{\mathcal{H}}$, $\epsilon_{\mathcal{H}}$ is negligible.

Oracle Collision Resistance. We say a chameleon hash function is $(\epsilon_{\mathcal{H}}, T_{\mathcal{H}})$-oracle collision resistant (OCR) if any adversary $\mathcal{A}$ without access to the trapdoor td, the success probability of finding a pair of collisions is at most $\epsilon_{\mathcal{H}}$ within time $T_{\mathcal{H}}$ in the following experiment.

$$\begin{aligned}\mathrm{Succ}^{\mathsf{ocr}}_{\mathcal{CMH},\mathcal{A}} \stackrel{\mathsf{def}}{=} \Pr[&(hk, td) \leftarrow \mathsf{CMkg}(1^k); \bar{x} \leftarrow \{0,1\}^*; \bar{w} \leftarrow \mathcal{R};\\ &y \leftarrow \mathsf{CMhash}(hk, \bar{x}, \bar{w}); x \leftarrow \mathcal{A}(hk, y); w \leftarrow \mathsf{CMswch}(td, \bar{x}, \bar{w}, x);\\ &(x', w') \leftarrow \mathcal{A}(hk, x, w) : (x', w') \neq (x, w) \wedge y = \mathsf{CMhash}(hk, x', w')]\end{aligned}$$

We say a chameleon hash function is oracle collision resistant, if for polynomially bounded $T_{\mathcal{H}}$, $\epsilon_{\mathcal{H}}$ is negligible.

Discussions. Oracle collision resistance has not been formally discussed before. However, it seems a very natural definition for chameleon hash functions, since it considers a collision cannot be found even after the adversary gets some hint from a switch oracle. Simultaneously, compared with the game defining collision resistance, the adversary seems to have quite limit power in choosing its target, since it is required to find collisions on a specified random hash value. At present, we don't know whether there are implications or separations between the above two notions. However, there are practical implementations of such chameleon hash functions provably secure under proper assumptions. We provide such an example here: A chameleon hash by combining Pedersen commitment with a normal collision resistant hash function appeared in [23], and it has been proven to be collision resistant under the discrete log assumption [23]. But one can easily come with a proof that it is also oracle collusion resistant under one-more discrete log assumption [26], which is equivalent to discrete log assumption in the generic group model [29]. We omit the details here.

Chameleon Hash as One-Time Signature. A strongly existentially unforgeable [3] one-time signature can be derived from an oracle collision resistant chameleon hash function. For key generation, run $(hk, td) \leftarrow \mathsf{CMkg}(1^k)$, select random $(\bar{x}, \bar{w})$ from corresponding spaces and compute $y = \mathsf{CMhash}(hk, \bar{x}, \bar{w})$. The verification key is $vk = (hk, y)$ and the signing key is $sigk = (td, \bar{x}, \bar{w})$. For signing, on message m, compute $s \leftarrow \mathsf{CMswch}(td, \bar{x}, \bar{w}, m)$, the signature on m is s. The correctness of the construction is easily verified.

To see strong unforgeability, a forger, given vk, will try to output (m', s'), such that $(m', s') \neq (m, s)$, where s is the signature on m chosen by the forger, and (m', s') a valid message/signature pair under vk. Then if $y = \mathsf{CMhash}(hk, m', s')$, a collision occurs for (m, s), which is against the assumption of oracle collision resistance. We then conclude the above signature scheme is strongly existentially unforgeable against (exactly one-time) adaptive chosen message attack.

2.6 Hard Problems

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties: (i) *Bilinearity*: For all $x, y \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}$, $e(x^a, y^b) = e(x, y)^{ab}$. (ii) *Non-degeneracy*: $e(g, g) \neq 1$. (iii) *Computability*: There is an efficient algorithm to compute $e(x, y)$ for any $x, y \in \mathbb{G}_1$. We review some hard problems related to bilinear maps: the decision bilinear Diffie-Hellman (DBDH) problem, the decision bilinear Diffie-Hellman inversion (DBDHI) problem, the decision linear (DLIN) problem and the decision augmented bilinear Diffie-Hellman exponent (DABDHE) problem. We say an assumption holds, if the advantage ϵ of any probabilistic polynomial bounded algorithm is negligible for the corresponding problem.

DBDH Problem. We say that the DBDH problem is (ϵ, T)-hard in $(\mathbb{G}_1, \mathbb{G}_2)$, if given 5-tuple $(g, g^a, g^b, g^c, w) \in (\mathbb{G}_1)^4 \times \mathbb{G}_2$ as input, any randomized algorithm $\mathcal{A}$ with running time at most T, distinguishes the BDH-tuple from a random tuple with advantage at most ϵ.

$$\mathrm{Adv}^{\mathsf{dbdh}}_{\mathbb{G}_1, \mathbb{G}_2, \mathcal{A}} \stackrel{\mathsf{def}}{=} \left|\Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, w) = 0]\right|$$

DBDHI Problem. We say that the DBDHI problem is (ϵ, q, T)-hard in $(\mathbb{G}_1, \mathbb{G}_2)$, if given $(q+2)$-tuple $(g, g^x, g^{x^2}, ..., g^{x^q}, w) \in (\mathbb{G}_1)^{q+1} \times \mathbb{G}_2$, where $x \in \mathbb{Z}_p$ as input, any randomized algorithm $\mathcal{A}$ with running time at most T, decides whether $w = e(g, g)^{1/x}$ with advantage at most ϵ.

$$\mathrm{Adv}^{\mathsf{dbdh}}_{\mathbb{G}_1, \mathcal{A}} \stackrel{\mathsf{def}}{=} |\Pr[\mathcal{A}(g, g^x, g^{x^2}, ..., g^{x^q}, e(g, g)^{1/x}) = 0] - \Pr[\mathcal{A}(g, g^x, g^{x^2}, ..., g^{x^q}, w) = 0]|$$

DLIN Problem. We say that the DLIN problem is (ϵ, T)-hard in $\mathbb{G}_1$, if given 6-tuple $(g_1, g_2, g_1^{r_1}, g_2^{r_2}, z, w) \in (\mathbb{G}_1)^6$ as input, where $(r_1, r_2) \in (\mathbb{Z}_p)^2$, any randomized algorithm $\mathcal{A}$ with running time at most T, decides whether $w = z^{r_1 + r_2}$ with advantage at most ϵ.

$$\mathrm{Adv}^{\mathsf{dbdh}}_{\mathbb{G}_1, \mathcal{A}} \stackrel{\mathsf{def}}{=} \left|\Pr[\mathcal{A}(g_1, g_2, g_1^{r_1}, g_2^{r_2}, z, z^{r_1 + r_2}) = 0] - \Pr[\mathcal{A}(g_1, g_2, g_1^{r_1}, g_2^{r_2}, z, w) = 0]\right|$$

It is believed that DLIN problem is hard even in a bilinear group pair where pairing is efficiently computable and its security can be proven in the generic group model.

DABDHE Problem. We say that the DABDHE problem is (ϵ, q, T)-hard in $(\mathbb{G}_1, \mathbb{G}_2)$, if given $(q+4)$-tuple $(g_1, g_2, g_1^x, ..., g_1^{x^q}, g_1^{x^{q+2}}, w) \in (\mathbb{G}_1)^{q+3} \times \mathbb{G}_2$, where $x \in \mathbb{Z}_p$ as input, any randomized algorithm $\mathcal{A}$ with running time at most T, decides whether $w = e(g_1, g_2)^{q+1}$ with advantage at most ϵ.

$$\mathrm{Adv}^{\mathsf{dabdhe}}_{\mathbb{G}_1,\mathbb{G}_2,\mathcal{A}} \stackrel{\mathsf{def}}{=} |\Pr[\mathcal{A}(g_1, g_2, g_1^x, ..., g_1^{x^q}, g_1^{x^{q+2}}, e(g_1, g_2)^{x^{q+1}}) = 0] - \Pr[\mathcal{A}(g_1, g_2, g_1^x, ..., g_1^{x^q}, g_1^{x^{q+2}}, w) = 0]|$$

3 Separable TBE/IBE to PKE Transforms

3.1 The Transforms

We give Transform T1 that achieves chosen ciphertext security from separable tag based primitives in Figure 1, then analyze its security in Theorem 1.

Theorem 1. *The public key encryption acquired via* T1 *is* $(\epsilon + \epsilon_{\mathcal{H}}, q, T + T_{\mathcal{H}} + O(qk))$-$\mathsf{IND\text{-}CCA}$*-secure assuming the separable tag based encryption is* (ϵ, q, T)-$\mathsf{IND\text{-}sTag\text{-}CCA}$*-secure and the collision resistant chameleon hash function is* $(\epsilon_{\mathcal{H}}, T_{\mathcal{H}})$*-collision resistant.*

Proof. We show how to build an adversary $\mathcal{B}$ breaks either the TBE or the chameleon hash by interacting with a PKE adversary. Denote $\langle u^*, v^*, r_2^* \rangle$ as the challenge ciphertext for $\mathcal{A}$. We distinguish two types of adversaries:

Type 1: For any valid decryption query $\langle u^{(i)}, v^{(i)}, r_2^{(i)} \rangle$, where $1 \leq i \leq q$, there is $(u^{(i)}, r_2^{(i)}, t^{(i)}) \neq (u^*, r_2^*, t^*)$, where $t^{(i)} = \mathsf{CMhash}(hk, u^{(i)}, r_2^{(i)})$ and $t^* = \mathsf{CMhash}(hk, u^*, r_2^*)$. We construct an adversary that breaks the TBE.

Type 2: There is at least one valid decryption query $\langle u^{(i)}, r_2^{(i)}, t^{(i)} \rangle = \langle u^*, r_2^*, t^* \rangle$ for some $1 \leq i \leq q$. We construct an adversary that breaks the collision resistance of chameleon hash.

Type 1 Adversary: Define $\mathcal{B}$ as follows:

Setup: $\mathcal{B}$ runs $(hk, td) \leftarrow \mathsf{CMkg}(1^k)$. Let $u' = f_1(pk, m', r_1')$, where m' is a dummy message and r_1' is random coin for TBEenc. $\mathcal{B}$ computes $t^* = \mathsf{CMhash}(hk, u', r_2')$, where r_2' is an auxiliary random coin for chameleon hash. $\mathcal{B}$ then submits t^* to its own challenger as the tag to be challenged. After receiving public key pk' from its challenger, $\mathcal{B}$ sets $pk = (pk', hk)$ and sends pk as the public key to a PKE adversary $\mathcal{A}$.

Transform: T1

$\mathsf{PKEkg}(1^k)$:	$\mathsf{PKEenc}(pk, m)$:	$\mathsf{PKEdec}(sk, c)$:
$(pk', sk') \leftarrow \mathsf{TBEkg}(1^k)$	$pk = \langle pk', hk \rangle$	$c = \langle u, v, r_2 \rangle$
$hk \leftarrow \mathsf{CMkg}(1^k)$	$r_1 \leftarrow \mathcal{R}_1$	$t \leftarrow \mathsf{CMhash}(hk, u, r_2)$
$pk \leftarrow \langle pk', hk \rangle$	$r_2 \leftarrow \mathcal{R}_2$	$m \leftarrow \mathsf{TBEdec}(sk, t, u\|\|v)$
$sk \leftarrow sk'$	$u \leftarrow f_1(pk', m, r_1)$	return m
return (pk, sk)	$t \leftarrow \mathsf{CMhash}(hk, u, r_2)$	
	$v \leftarrow f_2(pk', t, r_1)$	
	$c \leftarrow \langle u, v, r_2 \rangle$	
	return c	

† The trapdoor for chameleon hash function can be erased since it is not used elsewhere. $\mathcal{R}_1$ and $\mathcal{R}_2$ are corresponding spaces of random coins for TBEenc and CMhash.

Fig. 1. The Separable TBE to PKE Transform

Encryption Query: When $\mathcal{B}$ receives from $\mathcal{A}$ a pair of plaintexts (m_0, m_1) that $\mathcal{A}$ wants to be challenged on, $\mathcal{B}$ forwards (m_0, m_1) to its own challenge oracle. After receiving its challenge ciphertext $\langle u^*, v^* \rangle$ (under tag t^*), $\mathcal{B}$ computes $r_2^* \leftarrow \mathsf{CMswch}(td, u', r_2', u^*)$. $\mathcal{B}$ then sends $c^* = \langle u^*, v^*, r_2^* \rangle$ to $\mathcal{A}$ as the challenge ciphertext. Due to the uniformness of the chameleon hash, the distribution of the challenge is exactly as a real attack.

Decryption Queries: For decryption query $c = \langle u^{(i)}, v^{(i)}, r_2^{(i)} \rangle$, $\mathcal{B}$ checks whether $(u^{(i)}, r_2^{(i)}) = (u^*, r_2^*)$ and $v^{(i)} \neq v^*$. If yes, this is an invalid ciphertext and $\mathcal{B}$ rejects. Otherwise, $\mathcal{B}$ sends $\langle u^{(i)}, v^{(i)}, t^{(i)} \rangle$ to its own decryption oracle and forwards to $\mathcal{A}$ whatever its decryption oracle replies.

Guess: When $\mathcal{A}$ outputs a guess b' on c^*, $\mathcal{B}$ outputs the same bit as its answer.

From the description of $\mathcal{B}$, it is easily verified that the key generation is simulated perfectly. Furthermore, because of the uniformness property, r_2^* is uniformly distributed, thus the challenge oracle is also perfectly simulated. Finally, if $\mathcal{A}$ succeeds, $\mathcal{B}$ also succeeds.

Type 2 adversary: For Type 2 adversary has many similarities with Type 1, so we only give the sketch. For setup, $\mathcal{B}$ receives hk from its challenger. $\mathcal{B}$ then generates $(pk', sk') \leftarrow \mathsf{TBEkg}(1^k)$ and sets the public key as (pk', hk). $\mathcal{B}$ keeps sk' as the secret key. Since $\mathcal{B}$ has sk', all decryption queries can be answered perfectly. For challenge, upon receiving (m_0, m_1) from $\mathcal{A}$, $\mathcal{B}$ first picks $b \leftarrow \{0, 1\}$ and sets $u^* = f_1(pk', m_b, r)$, where r is chosen uniformly from corresponding randomness space. $\mathcal{B}$ then outputs u^* to its hash challenger. After receiving r_2^* from the challenger, $\mathcal{B}$ sets $t^* \leftarrow \mathsf{CMhash}(hk, u^*, r_2^*)$ and $v^* = f_2(pk', t^*, r)$, and sends $\langle u^*, v^*, r_2^* \rangle$ to $\mathcal{A}$ as the challenge ciphertext. One can verify this is a valid challenge. Finally, when decryption query $\langle u^{(i)}, v^{(i)}, r_2^{(i)} \rangle$ is queried for some i, where $\mathsf{CMhash}(hk, u^{(i)}, r_2^{(i)}) = t^*$ and $(u^{(i)}, r_2^{(i)}) \neq (u^*, r_2^*)$, $\mathcal{B}$ outputs $(u^{(i)}, r_2^{(i)})$ as a collision for its challenger. We conclude $\mathcal{B}$ break collision resistance with the same probability as $\mathcal{A}$'s advantage in guessing b.

Summarizing two cases, we have the claimed results. □

We further present another transform (T1′) based on IBE in Figure 2. Since an (ϵ, q, T)-IND-sID-CPA-Secure identity based encryption implies an (ϵ, q, T)-IND-sTag-CCA-secure tag based encryption, we have an immediate corollary for the security of this transform.

Transform: T1′		
$\mathsf{PKEkg}(1^k)$:	$\mathsf{PKEenc}(pk, m)$:	$\mathsf{PKEdec}(sk, c)$:
$(params, msk) \leftarrow \mathsf{IBEkg}(1^k)$	$pk = \langle params, hk \rangle$	$c = \langle u, v, r_2 \rangle$
$hk \leftarrow \mathsf{CMkg}(1^k)$	$r_1 \leftarrow \mathcal{R}_1$	$id \leftarrow \mathsf{CMhash}(hk, u, r_2)$
$pk \leftarrow \langle params, hk \rangle$	$r_2 \leftarrow \mathcal{R}_2$	$sk_{id} \leftarrow \mathsf{IBEext}(sk, id)$
$sk \leftarrow msk$	$u \leftarrow f_1(params, m, r_1)$	$m \leftarrow \mathsf{IBEdec}(sk_{id}, id, u\|\|v)$
return (pk, sk)	$t \leftarrow \mathsf{CMhash}(hk, u, r_2)$	return m
	$v \leftarrow f_2(params, id, r_1)$	
	$c \leftarrow \langle u, v, r_2 \rangle$	
	return c	

† Again, td can be erased. $\mathcal{R}_1$ and $\mathcal{R}_2$ are corresponding spaces of random coins for IBEenc and CMhash.

Fig. 2. The Separable IBE to PKE Transform

Corollary 1. *The public key encryption acquired via* T1′ *is* $(\epsilon + \epsilon_{\mathcal{H}}, q, T + T_{\mathcal{H}} + O(qk))$-IND-CCA-*secure assuming the separable identity based encryption is* (ϵ, q, T)-IND-sID-CPA-*Secure and the collision resistant chameleon hash function is* $(\epsilon_{\mathcal{H}}, T_{\mathcal{H}})$-*collision resistant.*

3.2 Further Improvements

Note that the performance of T1 and T1′ can be further optimized by replacing chameleon hash functions using practical hash functions (e.g. SHA-1, or SHA-2 for higher security). We have to modify our assumptions and corresponding proofs. Instead of possessing td of a chameleon hash function, the simulator $\mathcal{B}$ is given one chance of accessing a collision oracle (possibly inefficient) that finds collision on the hash function. The uniformness of such hash functions should also be rephrased accordingly. With these changes, it is still possible to prove the security of the modified constructions according to previous strategies. The details are omitted here.

4 A Generic TBE to PKE Transform

We give the description of a generic TBE to PKE transform using chameleon hash in Figure 3, and naturally it can be rewritten to fit IBE case accordingly. The transform mimics [13] with a one-time signature replaced by a chameleon hash function that is discussed in Section 2.5. This transform shows another tradeoff between computational cost and ciphertext size for generic TBE/IBE

Transform: T2		
$\mathsf{PKEkg}(1^k)$: $(pk, sk) \leftarrow \mathsf{TBEkg}(1^k)$ return (pk, sk)	$\mathsf{PKEenc}(pk, m)$: $(hk, sk) \leftarrow \mathsf{CMkg}(1^k)$ $\bar{u} \leftarrow \mathcal{C}$ $\bar{r} \leftarrow \mathcal{R}$ $t' \leftarrow \mathsf{CMhash}(hk, \bar{u}, \bar{r})$ $t \leftarrow hk\|t'$ $u \leftarrow \mathsf{TBEenc}(pk, t, m)$ $r \leftarrow \mathsf{CMswch}(td, \bar{u}, \bar{r}, u)$ $c \leftarrow \langle u, t, r \rangle$ return c	$\mathsf{PKEdec}(sk, c)$: $c = \langle u, hk\|t', r \rangle$ test if $t' \stackrel{?}{=} \mathsf{CMhash}(hk, u, r)$ if invalid, return "$\perp$" $m \leftarrow \mathsf{TBEdec}(sk, t, u)$ return m

† $\mathcal{C}$ and $\mathcal{R}$ are corresponding spaces of ciphertext and random coins of CMhash respectively.

Fig. 3. A Generic TBE to PKE Transform Using Chameleon Hash

transforms. It is not hard to come up with an IBE version of T2, just as we did previously to T1. We omit the details here. Finally, Theorem 2 guarantees the security of T2.

Theorem 2. *The public key encryption acquired via* T2 *is* $(\epsilon + \epsilon_{\mathcal{H}}, q, T + T_{\mathcal{H}} + O(qk))$-IND-CCA-*secure assuming the tag based encryption is* (ϵ, q, T)-IND-sTag-CCA-*secure and the chameleon hash function is* $(\epsilon_{\mathcal{H}}, T_{\mathcal{H}})$-*oracle collision resistant.*

Proof Sketch. The idea of the proof is quite similar to [13]. Let $\mathcal{A}$ be a PKE adversary. Denote $\langle u^{(i)}, t^{(i)}, r^{(i)} \rangle$, where $1 \leq i \leq q$, as decryption queries by $\mathcal{A}$. Denote $\langle u^*, t^*, r^* \rangle$ be the challenge ciphertext for $\mathcal{A}$. We consider two types of adversaries.

Type 1: For any $\langle u^{(i)}, t^{(i)}, r^{(i)} \rangle$, where $t^{(i)} \neq t^*$, we build an adversary $\mathcal{B}$ against the underlying TBE.

Type 2: There exists a query $\langle u^{(i)}, t^*, r^{(i)} \rangle$, where $(u^{(i)}, r^{(i)}) \neq (u^*, r^*)$, $t^* = hk^*\|t'^*$ and $t'^* = \mathsf{CMhash}(hk^*, u^{(i)}, r^{(i)})$. We construct an adversary $\mathcal{B}$ against the oracle collision resistance of the underlying chameleon hash.

Type 1 Adversary. For setup, $\mathcal{B}$ runs $(hk, td) \leftarrow \mathsf{CMkg}(1^k)$, chooses dummy $\bar{u}$ and random coin $\bar{r}$ and sets $t' \leftarrow \mathsf{CMhash}(hk, \bar{u}, \bar{r})$. Then $\mathcal{B}$ outputs $t^* = (hk\|t')$ as the selective tag for a TBE challenger. After receiving pk from the TBE challenger, $\mathcal{B}$ forwards pk to $\mathcal{A}$. For decryption, since there is no query with tag $t^{(i)} = t^*$, all decryption queries of $\mathcal{A}$ can be forwarded to the TBE challenger and answered perfectly. For challenge, after $\mathcal{A}$ submits a pair of chosen message (m_0, m_1), $\mathcal{B}$ forwards to the TBE challenger and receives a challenge ciphertext u^*. $\mathcal{B}$ then computes $r^* \leftarrow \mathsf{CMswch}(td, \bar{u}, \bar{r}, u^*)$, and gives $\langle u^*, t^*, r^* \rangle$ to $\mathcal{A}$ as the challenge. Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs as its guess. From above descriptions we can see that $\mathcal{B}$ succeeds with exactly the probability of $\mathcal{A}$.

Type 2 Adversary. For setup, $\mathcal{B}$ runs $(pk, sk) \leftarrow \mathsf{TBEkg}(1^k)$. $\mathcal{B}$ then forwards pk to $\mathcal{A}$ as the public key of the PKE. Additionally, on $\mathcal{B}$'s request, a chameleon hash challenger runs $(hk^*, td^*) \leftarrow \mathsf{CMkg}(1^k)$ and computes $t'^* \leftarrow \mathsf{CMhash}(hk^*, \bar{u}, \bar{r})$, where $\bar{u}$ and $\bar{r}$ are uniformly sampled from corresponding spaces. Then $t^* = hk||t'^*$ is given to $\mathcal{B}$. For decryption, since $\mathcal{B}$ knows sk, all decryption queries will be handled perfectly. For challenge, $\mathcal{B}$ receives (m_0, m_1) from $\mathcal{A}$ and sets $u^* \leftarrow \mathsf{TBEenc}(pk, t^*, m_b)$ for $b \leftarrow \{0, 1\}$. $\mathcal{B}$ submits u^* to the chameleon hash challenger, who computes $r^* \leftarrow \mathsf{CMswch}(td^*, \bar{u}, \bar{r}, u^*)$ and returns r^* to $\mathcal{B}$. $\mathcal{B}$ then gives $\langle u^*, t^*, r^* \rangle$ to $\mathcal{A}$ as a challenge. After receiving a ciphertext query of the form $\langle u^{(i)}, t^*, r^{(i)} \rangle$ where $t'^* = \mathsf{CMhash}(hk^*, u^{(i)}, r^{(i)})$ and $(u^{(i)}, r^{(i)}) \neq (u^*, r^*)$, $\mathcal{B}$ outputs $(u^{(i)}, r^{(i)})$ as a collusion for the chameleon hash. It is verified that $\mathcal{B}$ breaks the oracle collision resistance of chameleon hash at exactly $\mathcal{A}$'s advantage in correctly guess b.

Summarizing the above two cases we prove the claim. □

Although additional computational cost may be involved in the key generation and evaluation of the chameleon hash, it can be improved by pre-computation. It is worth repeating that oracle collision chameleon hash functions can be built from many number theoretic assumptions, and the public verifiability of underlying primitives is preserved by using T2.

5 Applications

5.1 Practical CCA-Secure PKE Schemes

Our methods achieves CCA-security with tight reductions to underlying selective-Tag based primitives. Instantiate T1 with IND-sTag-CCA-secure TBE, one gets more efficient scheme with public verifiability. We give such a scheme based on Kiltz TBE [22]. To remark, the Cramer-Shoup encryption [15] can be viewed as applying the BMW transform [11] to a related TBE with adaptive chosen tag security. Finally, instantiate T1′ with Gentry IBE, one gets an efficient PKE based on DABDHE assumption.

The Scheme		
$\mathsf{PKEkg}(1^k)$:	$\mathsf{PKEenc}(pk, m)$:	$\mathsf{PKEdec}(sk, c)$:
$g_1, h \leftarrow \mathbb{G}_1$	$r_1, r_2, r_3 \leftarrow \mathbb{Z}_p^*$	$c = \langle u_1, u_2, e, v_1, v_2, r_3 \rangle$
$x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}_p^*$	$u_1 \leftarrow g_1^{r_1}$	$t \leftarrow H(g_1^{H(u_1,u_2,e)} h^{r_3})$
$g_2, z \leftarrow \mathbb{G}_2$	$u_2 \leftarrow g_2^{r_2}$	test whether
s.t. $g_1^{x_1} = g_2^{x_2} = z$	$e \leftarrow M \cdot z^{r_1+r_2}$	$v_1 = u_1^{tx_1+y_1} \wedge v_2 = u_2^{tx_2+y_2}$
$u_1 \leftarrow g_1^{y_1}$	$t \leftarrow H_2(g_1^{H_1(u_1,u_2,e)} h^{r_3})$	if invalid, return "$\perp$"
$u_2 \leftarrow g_2^{y_2}$	$v_1 \leftarrow z^{tr_1} u_1^{r_1}$	otherwise $m \leftarrow \frac{e}{u_1^{x_1} u_2^{x_2}}$
choose $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$	$v_2 \leftarrow z^{tr_2} u_2^{r_2}$	return m
$pk \leftarrow (g_1, g_2, u_1, u_2, z, h, H)$	$c \leftarrow \langle u_1, u_2, e, v_1, v_2, r_3 \rangle$	
$sk \leftarrow (x_1, x_2, y_1, y_2)$	return c	
return (pk, sk)		

† $(\mathbb{G}_1, \mathbb{G}_2)$ is a bilinear group pair with prime order p. H_1, H_2 can be replaced by injective mappings.

Fig. 4. Secure PKE Based on DLIN Assumption

A Publicly Verifiable CCA-Secure PKE without Pairings. We instantiate our method with Kiltz TBE [22], and present an efficient IND-CCA-secure public key encryption scheme in Figure 4. We note that while Kiltz KEM [22] combined with a CCA-secure symmetric key encryption achieves better performance regarding chosen ciphertext security, but our scheme enjoys additionally a capability of threshold decryptions. Here we instantiate the chameleon hash by using a collision resistant hash function combined with the Pedersen commitment [23].

5.2 Extensions

CCA-Secure Hierarchical Identity Based Encryption. Since most IBEs in the standard model can be extended to the hierarchical IBE setting, our methods operate also on these HIBE schemes. Applying our transforms to an $(\ell+1)$-level semantically secure HIBE against chosen plaintext attack results in an ℓ-level CCA-secure HIBE with marginal computational cost, small ciphertext overhead and tight security reduction.

Table 1. Comparisons of Schemes

Scheme	Assumption	Ciphertext Overhead	Without Pairing?	Generic?	Public Verifiable?
KD	DDH	$2\|\mathbb{G}\| + \|\mathsf{Mac}\|$	yes	—	—
CHK/BB1	DBDH	$2\|\mathbb{G}_1\| + O(k^2)$	—	yes	yes
CHK/BB2	DBDHI	$2\|\mathbb{G}_1\| + O(k^2)$	—	yes	yes
BK/BB1	DBDH	$3\|\mathbb{G}_1\| + \|\mathsf{Mac}\|$	—	yes	—
BK/BB2	DBDHI	$3\|\mathbb{G}_1\| + \|\mathsf{Mac}\|$	—	yes	—
BK/Kiltz	DLIN	$5\|\mathbb{G}_1\| + \|\mathsf{Mac}\|$	yes	yes	—
Kiltz(KEM)	DLIN	$4\|\mathbb{G}_1\|$	yes	—	—
BMW/BB1	DBDH	$2\|\mathbb{G}_1\|$	—	—	—
BMW/Waters	DBDH	$2\|\mathbb{G}_1\|$	—	—	—
T1/Kiltz	DLIN	$4\|\mathbb{G}_1\| + \|r\|$	yes	—	yes
T1′/BB1	DBDH	$2\|\mathbb{G}_1\| + \|r\|$	—	—	yes
T1′/Gentry	DABDHE	$\|\mathbb{G}_1\|+\|\mathbb{G}_2\| + \|r\|$	—	—	yes
T2/BB1	DBDH	$2\|\mathbb{G}_1\| + \|hk\| + \|t\| + \|r\|$	—	yes	yes

† KD is Kurosawa-Desmedt [24]. Kiltz is Kiltz TBE [22]. Waters is Waters IBE [30]. Gentry is Gentry IBE [18]. All schemes in the table are PKEs, except that BMW/BB1 and Kiltz(KEM) [22] are CCA-secure KEMs. Such KEMs can combine with symmetric key encryption (SKE) (e.g. block ciphers run in CMC mode [20] or EME mode [21]) which has no overhead, however, such operations are usually computationally less efficient than passively secure SKEs combined with Macs. The cost of computing one-time signatures and symmetric key primitives are neglected here. p is the order of $\mathbb{G}_1$. $\mathbb{G}$ is a group where DDH problem is hard. $(\mathbb{G}_1, \mathbb{G}_2)$ is a bilinear group pair. One may assume $|\mathbb{G}| = 1024$ (or 160 by using elliptic curve) and $|\mathbb{G}_1| = 512$ and $|\mathbb{G}_2| = 1024$ for symmetric bilinear group. r is a random coin for chameleon hash functions and is chosen uniformly from $\mathbb{Z}_p$, where p is the order of $\mathbb{G}_1$. hk is a hash key, and t' is a hash value. For practical implementations of chameleon hash, the size of hk and t' can be further optimized by reusing the system parameters. One-time signatures based on number-theoretic assumptions are no better than T2.

Non-interactive CCA-Secure Threshold Decryption. If the underlying tag based primitive is publicly verifiable, one can easily build non-interactive threshold encryption against chosen ciphertext attack with our method. It is quite natural if one follows previous work, e.g., [11], and the details and concrete instantiations are omitted here. We note that alternatively, this can be achieved by building a tag-KEM [2], combined with a semantically secure DEM. However, one may need to take care on the security of underlying primitives to build the tag-KEM. A recent work [1] addresses this in more details.

6 Comparisons

We compare some typical PKE schemes in Table 1. Note that our transforms can be further improved according to the discussions in Section 3.2, thus we come to conclusion that our transforms are efficient and widely applicable.

Acknowledgement

We thank Nuttapong Attrapadung, Jun Furukawa, Goichiro Hanaoka and Isamu Teranishi for helpful discussions. We thank Masayuki Abe and Eike Kiltz for pointing out the existence of Kiltz KEM and helping us discover unprecise presentations of Table 1 in a previous draft. We also thank anonymous referees of ACNS'07 for many helpful comments.

References

1. M. Abe, Y. Cui, H. Imai, and E. Kiltz. Efficient Hybrid Encryption from ID-Based Encryption. Cryptology ePrint Archive, `http://eprint.iacr.org/2007/023/`, 2007.
2. M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A New Framework for Hybrid Encryption. Cryptology ePrint Archive, `http://eprint.iacr.org/2005/027/`, 2005.
3. J.H. An, Y. Dodis, and Tal Rabin. On the Security of Joint Signature and Encryption. In *Eurocrypt'02*, volume 2332 of *LNCS*, pages 83–107. Springer, 2002.
4. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *CRYPTO'98*, volume 1462 of *LNCS*. Springer, 1998.
5. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pages 62–73. ACM Press, 1993.
6. D. Boneh and X. Boyen. Efficient Selective-ID Identity Based Encryption without Random Oracles. In *EUROCRYPT'04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
7. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO'04*, volume 3152 of *LNCS*, pages 443–459, 2004.
8. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security From Identity-Based Encryption. *SIAM Journal on Computing*, 36(5):915–942, 2006.
9. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO'01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

10. D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In *CT-RSA'05*, volume 3376, pages 87–103. Springer, 2005.
11. X. Boyen, Q. Mei, and B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. In *ACM CCS'05*, pages 320–329. ACM Press, 2005.
12. R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *EUROCRYPT'03*, volume 2656 of *LNCS*, pages 255–273. Springer, 2003.
13. R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT'04*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
14. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In *the 8th IMA international conference on cryptography and coding*, volume 2260 of *LNCS*, pages 360–363. Springer, 2001.
15. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.
16. R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Chosen Ciphertext Secure Public Key Encryption . In *EUROCRYPT'02*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.
17. D. Dolev, C. Dwork, and M. Naor. Nonmalleable Cryptography. In *STOC'91*, pages 542–552. ACM, 1991. Full version in SIAM Journal on Computing, 30(2):391-437, 2000.
18. C. Gentry. Practical Identity-Based Encryption Without Random Oracles. In *EUROCRYPT'06*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.
19. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
20. S. Halevi and P. Rogaway. A Parallelizable Enciphering Mode. In *Crypto'03*, volume 2729 of *LNCS*, pages 482–499. Springer, 2003.
21. S. Halevi and P. Rogaway. A Parallelizable Enciphering Mode. In *CT-RSA'04*, volume 2964 of *LNCS*, pages 292–304. Springer, 2004.
22. E. Kiltz. Chosen-Ciphertext Security from Tag-Based Encryption. In *TCC'06*, volume 1070 of *LNCS*, pages 581–600. Springer, 2006.
23. H. Krawczyk and T. Rabin. Chameleon Hashing and Signatures. 1997. Cryptology ePrint Archive Report. Available at `http://eprint.iacr.org/1998/010`.
24. K. Kurosawa and Y. Desmedt. A New Paradigm of Hybrid Encryption Scheme. In *CRYPTO'04*, volume 3152 of *LNCS*, pages 426–442. Springer, 2004.
25. M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC'90*, pages 427–437. ACM Press, 1990.
26. P. Paillier and D. Vergnaud. Discrete-Log-Based Signatures May Not Be Equivalent to Discrete Log. In *Asiacrypt'05*, volume 3788 of *LNCS*, pages 1–20. Springer, 2005.
27. C. Rackoff and D.R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.
28. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS'99*, pages 543–553. IEEE Computer Society, 1999.
29. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, 1997.
30. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *EUROCRYPT'05*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.