Chosen-Ciphertext Security from Tag-Based Encryption

Eike Kiltz[1]

CWI Amsterdam**
The Netherlands
kiltz@cwi.nl
http://kiltz.net

Abstract. One of the celebrated applications of Identity-Based Encryption (IBE) is the Canetti, Halevi, and Katz (CHK) transformation from any (selective-identity secure) IBE scheme into a full chosen-ciphertext secure encryption scheme. Since such IBE schemes in the standard model are known from previous work this immediately provides new chosen-ciphertext secure encryption schemes in the standard model.
This paper revisits the notion of Tag-Based Encryption (TBE) and provides security definitions for the selective-tag case. Even though TBE schemes belong to a more general class of cryptographic schemes than IBE, we observe that (selective-tag secure) TBE is a sufficient primitive for the CHK transformation and therefore implies chosen-ciphertext secure encryption.
We construct efficient and practical TBE schemes and give tight security reductions in the standard model from the Decisional Linear Assumption in gap-groups. In contrast to all known IBE schemes our TBE construction does not directly deploy pairings. Instantiating the CHK transformation with our TBE scheme results in an encryption scheme whose decryption can be carried out in one single multi-exponentiation.
Furthermore, we show how to apply the techniques gained from the TBE construction to directly design a new Key Encapsulation Mechanism. Since in this case we can avoid the CHK transformation the scheme results in improved efficiency.

1 Introduction

Since Diffie and Hellman proposed the idea of public key cryptography [14], one of the most active area of research in the field has been the design and analysis of public key encryption (PKE) schemes. In [16, 27] efficient primitives were suggested from which to build encryption schemes. Formal models of security were developed in [19, 23, 26] and nowadays it is widely accepted that security against chosen-ciphertext attacks provides the "right level of security" for public-key encryption schemes.

** The paper was written while the author was a visitor at University of California, San Diego, supported by a DAAD postdoc fellowship.

There have been numerous efficient schemes that were shown to be chosen-ciphertext secure in the *random oracle model* [2]. Unfortunately a proof in the random oracle model can only serve as a heuristic argument and has proved to possibly lead to insecure schemes when the random oracles are implemented in the standard model (see, e.g., [10]).

Dolev, Dwork, and Naor [15] were the first to come up with a public-key encryption scheme provably chosen-ciphertext secure in the standard model (without random oracles). Later Cramer and Shoup [12] presented the first really practical public-key encryption scheme. Their approach was further generalized in [13] and later shown by Elkind and Sahai [17] to fit into a more general framework. The nowadays most efficient chosen-ciphertext secure encryption scheme in the standard model is the one due to Kurosawa and Desmedt [21, 1] itself being an improvement of the original Cramer-Shoup scheme. Both schemes, Cramer-Shoup and Kurosawa-Desmedt are secure under the Decisional Diffie-Hellman (DDH) assumption.

From IBE to PKE. One of the recent celebrated applications of Identity-Based Encryption (IBE) is the work due to Canetti, Halevi, and Katz [11] showing an elegant black-box transformation from any IBE into a PKE scheme without giving up its efficiency. We will refer to this as the *CHK transformation*. If the IBE scheme is *selective-identity* secure then the resulting PKE scheme is chosen-ciphertext secure. Efficient constructions of IBE schemes in the standard model were recently developed by Boneh and Boyen [3] so the CHK transformation provides further alternative instances of chosen-ciphertext secure PKE schemes in the standard model.[1]

Another fact worth mentioning about the CHK transformation is that it does not seem to fall into the general framework characterized by Elkind and Sahai. Boneh and Katz [7] later improve the CHK transformation resulting in shorter ciphertexts and more efficient encryption/decryption. Since the two IBE schemes from [3] employ pairing operations the resulting schemes are still less efficient than the Kurosawa-Desmedt scheme.

Tag-Based Encryption. MacKenzie, Reiter, and Yang [22] introduce the notion of tag-based encryption (TBE) and show (independent from [11]) that the CHK transformation also transforms any "weakly secure" TBE scheme into a chosen-ciphertext secure PKE scheme. However, the only TBE schemes in the standard model mentioned in [11] are directly derived from known PKE schemes (for example the Cramer-Shoup scheme) and the CHK transformation applied to TBE schemes does not readily give us new instantiations of chosen-ciphertext secure PKE schemes.

[1] The underlying computational assumptions for the security reduction of the two IBE schemes from [3] are both "pairing-assumptions", i.e. the Bilinear Decisional Diffie-Hellman (BDDH) assumption and the q-strong Decisional Bilinear Diffie-Hellman Inversion (q-strong BDDHI) assumption.

1.1 Our Contribution

From TBE to PKE. As pointed out in the last two paragraphs selective-identity secure IBE (or weakly secure TBE) schemes are sufficient to construct chosen-ciphertext secure PKE schemes. The natural question that arises is if in the transformation some of the security requirements made to the IBE/TBE scheme can be dropped while still preserving security of the resulting PKE scheme. One of our contributions is to answer this question to the affirmative.

We revisit the security definitions for TBE schemes and introduce the notion of selective-tag secure TBE schemes. Selective-tag security for TBE can be seen as the selective-identity analog for IBE and is weaker than the TBE definition from [22] and the IBE definition from [11]. One of our main results is to show that selective-tag secure TBE is sufficient to build chosen-ciphertext secure PKE. Our construction uses the CHK transformations.

On the theoretical side our result underlines that for the CHK transformation, an IBE scheme is basically overkill since some of its functionality is superfluous. In particular, there is no need to have an IBE *key-derivation algorithm*, which seems to be what distinguishes IBE from all other public-key encryption primitives. The notion of TBE can be viewed as some sort of "flattened IBE scheme" (i.e., as IBE without key-derivation) and therefore exactly captures the above observation. Our contribution is to extract the best out of the afore mentioned papers: we are able to combine the known CHK transformation with a security requirement that is substantially weaker than the requirements that were believed to be necessary.

Comparing different security notions of TBE, IBE, and PKE. What distinguishes TBE from IBE is the IBE key-derivation algorithm. Indeed, as we will point out later, it seems to be hard to transform (even particular instances of) TBE schemes into IBE schemes. The difference between selective-tag TBE and weakly secure TBE schemes seems marginal at first glance but (similar to the IBE case [3]) it turns our that the "selective-tag" property is the key to make security proofs for TBE schemes much easier to construct. An even stronger security definition of TBE schemes was already used by Shoup [29] (where the tag was called "label"). Interestingly we show that such "strongly secure" TBE schemes are equivalent to chosen-ciphertext secure PKE schemes. Since the CHK transformation is black-box, our results imply that all the afore mentioned three flavors of TBE security together with chosen-ciphertext secure PKE are in fact all *equivalent* through efficient black-box reductions.

TBE and PKE are equivalent. So what is TBE good for? One may ask the question why to make the long detour over TBE when designing PKE schemes at all? The answer is simple. Since TBE is simpler and more general than PKE (and IBE) our hope is that TBE may prove itself useful in the future to come up with more chosen-ciphertext secure encryption in the standard model. In particular, we would like to have chosen-ciphertext secure PKE schemes based on different intractability assumptions. (Different from the BDDH or DDH assumption, hopefully even weaker or at least unrelated.)

An efficient TBE Scheme without Pairing Operations. To underline the usefulness of our TBE to PKE transformation we present an efficient TBE scheme that (in contrast to all known IBE schemes) does not directly rely on pairing operations for encryption and decryption. In particular, the decryption operation of our new TBE scheme is very efficient and (similar to the KD scheme) only performs one single multi-exponentiation. The recently introduced decisional linear (DLIN) assumption [4] states that, roughly, it should be computational infeasible to decide if $w = z^{r_1+r_2}$, given random $(g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w)$ as input. Our TBE scheme can be proved to meet the necessary security properties under the DLIN assumption in the standard model. The security reduction is tight, simple, and very intuitive. In contrast to all known efficient IBE schemes our TBE scheme does not directly use pairings. However, our proofs of security have to be carried out in *gap-groups* [25], i.e. groups in which CDH is believed to be hard even though they are equipped with an algorithm that efficiently solves the Decisional Diffie-Hellman (DDH) problem. One particular instance of such gap-groups (which is actually the only one we know at the time being) is obtained using pairings.

Instantiating the scheme with our TBE to PKE transformation we obtain a new and reasonably efficient chosen-ciphertext secure encryption scheme in the standard model based on the DLIN assumption. We remark that this is the first (practical) chosen-ciphertext secure PKE based on the DLIN assumption in the standard model.

Direct Key Encapsulation. A key encapsulation mechanism (KEM) is a light PKE scheme intended to encapsulate and decapsulate a random (symmetric) key. It is well known how to transform any chosen-ciphertext secure KEM into a fully fledged chosen-ciphertext secure PKE scheme using symmetric encryption (with appropriate security properties).

Surprisingly, our techniques from constructing the TBE scheme can also be exploited to directly build a chosen-ciphertext secure KEM in the standard model. Our construction avoids the CHK transformations and (similar to [12, 21]) only deploys a target collision-resistant hash function. As a result the ciphertext size of the scheme is more compact compared to the PKE scheme obtained using the above transformation. Furthermore encryption and decryption can be done more efficiently. Our KEM construction is practical and enjoys a simple proof of security with a tight reduction to the DLIN assumption in the standard model.

1.2 Related Work

Independent of our work, Boyen, Mei, and Waters [9] recently look at some specific PKE schemes obtained from the CHK transformation instantiated with the IBE schemes from [3, 30] and show how to make the resulting schemes more efficient (in terms of computation time and ciphertext length). In particular, they also come up with a practical chosen-ciphertext secure KEM (BMW-KEM) whose security is based on the BDDH assumption in the standard model.[2] Compared to our KEM, the BMW-KEM is based on bilinear pairings and therefore results in a less efficient decryption algorithm (one pairing and one exponentiation compared to one multi-exponentiation in our KEM). The BMW-KEM, however, is slightly more efficient in terms of encryption operations and comes with smaller ciphertexts. Compared to our KEM, the Kurosawa-Desmedt PKE scheme provides the same efficiency for decryption whereas it is more efficient for encryption. In Section 7.1 we discuss efficiency of all known encryption schemes in the standard model. Comparing the overall performance of all known encryption schemes in the standard model the Kurosawa-Desmedt scheme [21] can still be considered as the most efficient.

However, in contrast to the Kurosawa-Desmedt/Cramer-Shoup scheme, our KEM shares with the BMW-KEM the nice property that the validity (or consistency) of ciphertexts can be verified even without knowledge the the secret key. This observation was recently used in [9] to propose a threshold cryptosystem based on their BMW-KEM. With a similar idea and also based on the public validity test our KEM can also be used to build a threshold encryption scheme.

2 Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \stackrel{\$}{\leftarrow} S$ denotes the operation of picking an element s of S uniformly at random. Unless otherwise indicated, algorithms are randomized. "PT" stands for polynomial time and "PTA" for polynomial-time algorithm or adversary. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, and letting z be the output.

3 Definitions

In this section we formally introduce PKE and TBE schemes together with a security definition. We also give a parameter generating algorithm for bilinear groups and pairings and state our complexity assumptions.

3.1 Public-Key Encryption

An *public-key encryption* (PKE) scheme $\mathcal{PKE} = (\mathsf{PKEkg}, \mathsf{PKEenc}, \mathsf{PKEdec})$ consists of three polynomial time algorithms (PTAs). Via $(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{PKEkg}(1^k)$

[2] We note that the same scheme as in [9] was independently discovered during research for this paper. Since [9] is already published at the time of writing this extended abstract we decided not to include it here.

the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $C \stackrel{\$}{\leftarrow} \mathsf{PKEenc}(pk, M)$ a sender encrypts a message M under the public key pk to get a ciphertext; via $M \leftarrow \mathsf{PKEdec}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a message. Associated to the scheme is a message space MsgSp. For consistency, we require that for all $k \in \mathbb{N}$ and messages $M \in \mathsf{MsgSp}(k)$ we have $\Pr[\mathsf{PKEdec}(sk, \mathsf{PKEenc}(pk, M)) = M] = 1$, where the probability is taken over the coins of all the algorithms in the expression above.

Privacy. Privacy follows [26]. Let $\mathcal{PKE} = (\mathsf{PKEkg}, \mathsf{PKEenc}, \mathsf{PKEdec})$ be an PKE scheme with associated message space MsgSp. To an adversary $\mathcal{A}$ we associate the following experiment:

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\text{pke-cca}}_{\mathcal{PKE},\mathcal{A}}(k) \\
\quad (pk, sk) \stackrel{\$}{\leftarrow} \mathsf{PKEkg}(1^k) \\
\quad (M_0, M_1, st) \stackrel{\$}{\leftarrow} \mathcal{A}^{\textsc{Dec}(\cdot)}(\texttt{find}, pk) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\}\ ;\ C^* \stackrel{\$}{\leftarrow} \mathsf{PKEenc}(pk, M_b) \\
\quad b' \stackrel{\$}{\leftarrow} \mathcal{A}^{\textsc{Dec}(\cdot)}(\texttt{guess}, C^*, st) \\
\quad \text{If } b \neq b' \text{ then return 0 else return 1}
\end{array}
$$

where the oracle $\textsc{Dec}(C)$ returns $M \leftarrow \mathsf{PKEdec}(sk, C)$ with the restriction that in the `guess` phase adversary $\mathcal{A}$ is not allowed to query oracle $\textsc{Dec}(\cdot)$ for the target ciphertext C^*. Both challenge messages are required to be of the same size ($|M_0| = |M_1|$) and in the message space $\mathsf{MsgSp}(k)$. We define the advantage of $\mathcal{A}$ in the above experiment as

$$
\mathbf{Adv}^{\text{pke-cca}}_{\mathcal{PKE},\mathcal{A}}(k) \;=\; \left| \Pr\left[\mathbf{Exp}^{\text{pke-cca}}_{\mathcal{PKE},\mathcal{A}}(k) = 1 \right] - \frac{1}{2} \right| .
$$

PKE scheme $\mathcal{PKE}$ is said to be *secure against chosen ciphertext attacks* (CCA-secure) if the advantage function $\mathbf{Adv}^{\text{pke-cca}}_{\mathcal{PKE},\mathcal{A}}$ is a negligible function in k for all PTAs $\mathcal{A}$.

The weaker security notion of *security against chosen-plaintext attacks* (CPA-security) is obtained in the above security experiment when depriving adversary $\mathcal{A}$ of the the access to the decryption oracle.

3.2 Tag-based Encryption

Informally, in a tag-based encryption scheme [22], the encryption and decryption operations take an additional "tag". A tag is simply a binary string of *appropriate length*, and need not have any particular internal structure. We define security for tag-based encryption in manners analogous to security for standard encryption schemes. In particular, we define selective-tag security against chosen-ciphertext attacks. The selective-tag variant is reminiscent to the selective-identity variant of IBE schemes [11] and was not considered in [22].

More formally, a *tag-based encryption* (TBE) scheme $\mathcal{TBE} = (\mathsf{TBEkg}, \mathsf{TBEenc}, \mathsf{TBEdec})$ consists of three PTAs. Via $(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{TBEkg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $C \stackrel{\$}{\leftarrow}$

$\mathsf{TBEenc}(pk, t, M)$ a sender encrypts a message M with tag t to get a ciphertext; via $M \leftarrow \mathsf{TBEdec}(sk, t, C)$ the possessor of secret key sk decrypts ciphertext C to get back a message or the symbol $reject$. Note that the tag t must explicitly be provided as the input of the decryption algorithm and is usually not explicitly contained in the ciphertext. Associated to the scheme is a message space MsgSp. For consistency, we require that for all $k \in \mathbb{N}$, all tags t and messages $M \in \mathsf{MsgSp}(k)$ we have $\Pr[\mathsf{TBEdec}(sk, t, \mathsf{TBEenc}(pk, t, M)) = M] = 1$, where the probability is taken over the choice of $(pk, sk) \xleftarrow{\$} \mathsf{TBEkg}(1^k)$, and the coins of all the algorithms in the expression above.

Privacy. To an adversary $\mathcal{A}$ we associate the following experiment:

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\text{tbe-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k) \\
\quad (t^*, st_0) \xleftarrow{\$} \mathcal{A}(1^k, \mathtt{init}) \\
\quad (pk, sk) \xleftarrow{\$} \mathsf{TBEkg}(1^k) \\
\quad (M_0, M_1, st) \xleftarrow{\$} \mathcal{A}^{\mathrm{DEC}(\cdot,\cdot)}(\mathtt{find}, pk, st_0) \\
\quad b \xleftarrow{\$} \{0,1\}\ ;\ C^*_{tbe} \xleftarrow{\$} \mathsf{TBEenc}(pk, t^*, M_b) \\
\quad b' \xleftarrow{\$} \mathcal{A}^{\mathrm{DEC}(\cdot,\cdot)}(\mathtt{guess}, C^*_{tbe}, st) \\
\quad \text{If } b \neq b' \text{ then return 0 else return 1}
\end{array}
$$

where the oracle $\mathrm{DEC}(C, t)$ returns $M \leftarrow \mathsf{TBEdec}(sk, t, C)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle Dec for tag t^* (called *target tag*). Both messages must be of the same size ($|M_0| = |M_1|$) and in the message space $\mathsf{MsgSp}(k)$. We define the advantage of $\mathcal{A}$ in the above experiment as

$$
\mathbf{Adv}^{\text{tbe-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k) \;=\; \left| \Pr\left[\mathbf{Exp}^{\text{tbe-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k) = 1 \right] - \frac{1}{2} \right| .
$$

TBE scheme $\mathcal{TBE}$ is said to be *selective-tag weakly secure against chosen ciphertext attacks* if the advantage function is negligible for all PTAs $\mathcal{A}$.

In the security experiment adversary $\mathcal{A}$ is allowed to make decryption queries for any tag $t \neq t^*$, t^* being the tag the challenge ciphertext is created with. In particular, this includes queries for the target ciphertext C^*_{tbe} (when queried with a different tag $t \neq t^*$). In other words, the security notion offers chosen-ciphertext security for all tags $t \neq t^*$ and chosen-plaintext security for $t = t^*$. The target tag t^* has to be output by $\mathcal{A}$ before even seeing the public key. That means that a simulator may "tailor" the public-key to secure the scheme with respect to the above definition.

Discussion of different TBE variants. Tags in public-key encryption were already considered by Shoup [29] (and were called "labels") and later by MacKenzie, Reiter, and Yang [22]. While functionality is the same as in our definition, in terms of security there are small but crucial differences between the definitions given in the different papers. We recall the two TBE security variants from [29, 22] and point out the differences to our definition. Let C^*_{tbe} be the target ciphertext and t^* be the target tag selected by the adversary $\mathcal{A}$ in the security experiment.

- To obtain the notion of *weak CCA security* for TBE schemes (as considered in [22][3]) we modify the above security experiment in a way such that $\mathcal{A}$ does not have to commit to the target tag t^* in the beginning of the experiment. Instead, $\mathcal{A}$ is allowed to choose t^* at the end of its `find` stage, possibly depending on the public key and on its queries. Clearly, this is a stronger security requirement.
- To get (full) *CCA-security* (as considered in [29]), we further modify the security experiment (of weak CCA security) such that the adversary is allowed to ask any decryption query suspect to $(t, C_{tbe}) \neq (t^*, C_{tbe}^*)$. In particular this includes queries for the target tag t^* as long as $C_{tbe} \neq C_{tbe}^*$.

The differences between the different TBE security notions are summarized in the following table.

TBE security	Restriction to $\text{Dec}(t, C_{tbe})$ queries	Selective-tag?
(full) CCA [29]	$(t, C_{tbe}) \neq (t^*, C_{tbe}^*)$	no
weak CCA [22]	$t \neq t^*$	no
selective-tag weak CCA	$t \neq t^*$	yes

Clearly, the three definitions form a hierarchy of security notions, Shoup's CCA security being the strongest and our selective-tag weak CCA security being the weakest. We want to remark that selective-tag weak CCA security is strictly weaker than weak CCA security, i.e. there exists a TBE scheme that is selective-tag but not weakly CCA secure. (This can be shown by an example recently used in [18] to show a similar separation related to IBE schemes.)

Relation between TBE and PKE. It is easy to see that by identifying a message/tag pair (M, t) with a message $M||t$, any CCA-secure PKE scheme is also a CCA-secure TBE scheme. On the other hand, by identifying a message M with message/tag pair (M, t) (for an arbitrary tag t that is appended to the ciphertext in the plain) any CCA-secure TBE scheme can be used as a CCA-secure PKE scheme. Note that the same trick is not possible anymore if we weaken the security requirement to the TBE scheme to weak CCA security. (An adversary against the CCA security of the PKE scheme could query the decryption oracle for (C_{tbe}^*, t) for $t \neq t^*$ what would give it the plaintext M_b.) The above remarks show that the two notions of CCA-secure TBE and CCA-secure PKE can in fact be seen as equivalent. Fig. 1 in Section 4 is summarizing the relations between PKE and the different security flavors of TBE.

3.3 Identity Based Encryption

An identity based encryption (IBE) scheme can be viewed as a special kind of tag-based encryption scheme where the tag t is associated with an identity id. The difference is that an IBE scheme is equipped with an additional algorithm,

[3] Note that weak CCA-security for TBE schemes was called CCA-security in [22]. But for its relation to PKE schemes we prefer to refer to it as weak CCA-security. This should become clear later.

the key derivation algorithm KeyDer. On input of the secret key sk and an identity id, KeyDer generates a user secret key $usk[id]$ for identity id. This secret key allows the identity to decrypt all messages that were encrypted to identity id. In the terminology of TBE this means that $usk[t]$ is a "wild-card" to decrypt arbitrary ciphertexts that were encrypted with tag t, without knowing the secret key. A formal definition of IBE, together with a security model for (selective-identity) chosen-plaintext security, is given in the full version [20].

Relation between IBE and TBE. By the above it is easy to see that every IBE scheme can be transformed into a TBE scheme while maintaining its security properties. In the transformation TBE tag t is identified with IBE identity id. The key generation and encryption algorithms are the same. The TBE decryption algorithm first computes the secret key $usk[t]$ for "identity" t and then uses the public IBE decryption algorithm to recover the plaintext. It is easy to verify that if the IBE scheme is (selective-identity) CPA-secure then the TBE scheme is (selective-tag) weakly CCA-secure.[4] Furthermore, a CCA-secure IBE scheme translates to a CCA-secure TBE scheme. (See full version [20] for exact IBE security definitions.)

To the best of our knowledge it is not known how to generically transform a TBE scheme into an IBE scheme. This seems particularly difficult since it is not clear how, in general, the user secret key $usk[id]$ of the IBE scheme can be defined since in TBE there is no such concept as the "user secret key".

The above observations together with the discussion from Section 3.2 indicate that the class of selective-tag weakly CCA-secure TBE schemes is more general than the class of weakly CCA-secure TBE/selective-identity CPA-secure IBE schemes and gives furthermore hope that TBE schemes in the weak selective-tag model are easier to construct. Fig. 1 in Section 4 is summarizing the relations between TBE and IBE.

4 Chosen-Ciphertext Security from Tag-Based Encryption

Canetti, Halevi, and Katz [11] demonstrate how to transform any selective-identity CPA-secure IBE scheme into a CCA-secure PKE scheme by adding a one-time signature (we will refer to this as CHK transformation). Independent of [11], MacKenzie, Reiter, and Yang [22] exploit the same construction as [11] and describe how to convert any weakly CCA-secure TBE scheme into a CCA-secure PKE scheme. In this section we combine the above three papers [11, 22, 7] and show that a selevtice-tag weakly CCA-secure TBE scheme is sufficient to construct an CCA-secure PKE scheme. More precisely, we note that the CHK transformation may as well be instantiated with any TBE scheme (the PKE decryption algorithm needs to be adapted to the TBE definition). If the TBE scheme is selective-tag weakly CCA-secure then the resulting PKE scheme is

[4] Note that CCA security for TBE schemes naturally corresponds to CPA security for IBE schemes.

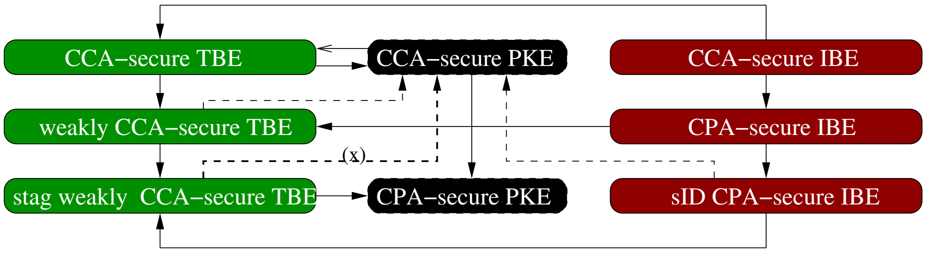

Fig. 1. Relation between IBE, TBE, and PKE with different security definitions. Solid arrows indicate direct implications, dashed lines indicate relations through a black-box reduction. All direct implications were discussed in Section 3. The upper left dashed black-box implication is due to [22], the right one due to [11], and the one with the marker (x) shows our contribution.

CCA-secure. We summarize the known relations among TBE, PKE, and IBE in Fig. 1. The results of this section settle the implication marked by (x).

4.1 The Transformation

Given a TBE scheme $\mathcal{TBE} = (\mathsf{TBEkg}, \mathsf{TBEenc}, \mathsf{TBEdec})$ with tag-space TagSp we construct a public-key encryption scheme $\mathcal{PKE} = (\mathsf{PKEkg}, \mathsf{PKEenc}, \mathsf{PKEdec})$. In the construction, we use a one-time signature scheme $\mathcal{OTS} = (\mathsf{SKG}, \mathsf{SIGN}, \mathsf{VFY})$ in which the verification key output by $\mathsf{SKG}(1^k)$ is an element from TagSp. We require that this scheme be secure in the sense of strong unforgeability (cf. [20]). The transformation defines the public/secret key pair of the PKE scheme to be the public/secret key pair of the TBE scheme, i.e. $\mathsf{PKEkg}(1^k)$ outputs whatever $\mathsf{TBEkg}(1^k)$ outputs. The construction proceeds as follows:

TBE to PKE transformation	
$\mathsf{PKEenc}(pk, M)$	$\mathsf{PKEdec}(sk, C)$
$(vk, sigk) \xleftarrow{\$} \mathsf{SKG}(1^k)$	Parse C as (C_{tbe}, vk, sig)
$C_{tbe} \xleftarrow{\$} \mathsf{TBEenc}(pk, vk, M)$	If $\mathsf{VFY}(vk, C_{tbe}, sig) = \mathtt{reject}$
$sig \xleftarrow{\$} \mathsf{SIGN}(sigk, C_{tbe})$	then return $\mathtt{reject}$.
Return $C \leftarrow (C_{tbe}, vk, sig)$	Else return $M \leftarrow \mathsf{TBEdec}(sk, vk, C_{tbe})$

It is easy to check that the above scheme satisfies correctness.

Let us now give some intuition why the PKE scheme is CCA-secure. Let (C^*_{tbe}, vk^*, sig^*) be the challenge ciphertext output by the simulator in the security experiment. It is clear that, without any decryption oracle queries, the value of the bit b remains hidden to the adversary. This is so because C^*_{tbe} is output by TBEenc which is CPA-secure, vk^* is independent of the message, and sig^* is the result of applying the one-time signing algorithm to C^*_{tbe}.

We claim that decryption oracle queries cannot further help the adversary in guessing the value of b. Consider an arbitrary ciphertext query $(C_{tbe}, vk, sig) \neq (C^*_{tbe}, vk^*, sig^*)$ made by the adversary during the experiment. If $vk = vk^*$ then

$(C_{tbe}, sig) \neq (C^*_{tbe}, sig^*)$ and the decryption oracle will answer `reject` since the adversary is unable to forge a new valid signature sig with respect to vk^*. If $vk \neq vk^*$ then the decryption query will not help the adversary since the actual decryption using $\mathcal{TBE}$ will be done with respect to a tag vk different to the target tag vk^*. A formalization of the above arguments leads to the following:

Theorem 1. *Assuming the TBE scheme is selective-tag chosen-ciphertext secure, the $\mathcal{OTS}$ is a strong, one-time signature scheme, then the above public-key encryption scheme is chosen-ciphertext secure.*

The security reduction is tight (linear) with respect to all the public-key components. The proof follows along the lines of [11, 5] and is therefore omitted here. We note that the CHK transformation can also be used to transform a (straight-forward definition of) tag-based KEM into a full KEM.

For simplicity we only described the CHK transformation in this Section. We want to remark that the more efficient BK transformation [7, 5] (which basically employs a MAC insteas of a signature) works as well for TBE schemes. The use of a MAC instead of a one-time signature somewhat complicates exposition and proof. The description of the BK transformation, together with all necessary definitions, is deferred to the full version [20].

5 An Efficient TBE scheme based on the Linear Assumption

In this section we demonstrate the usefulness of the TBE to PKE transformation of Section 4. Whereas the only known IBE schemes are using pairings [3] we give a simple and practical TBE scheme that does not perform any pairing operation.

5.1 Parameter generation algorithm for gap groups.

All schemes will be parameterized by a *gap parameter generator*. This is a PTA $\mathcal{G}$ that on input 1^k returns the description of an multiplicative cyclic group $\mathbb{G}$ of prime order p, where $2^k < p < 2^{k+1}$, and the description of a Diffie-Hellman oracle DDHvf. A tuple $(g, g^x, g^y, g^z) \in \mathbb{G}^4$ is called a *Diffie-Hellman tuple* if $xy = z \bmod p$. The oracle DDHvf is a PTA that for each input $(g, g^x, g^y, g^z) \in \mathbb{G}^4$ outputs 1 if (g, g^x, g^y, g^z) is a Diffie-Hellman tuple and 0 otherwise. More formally we require that for each $(\mathbb{G}, p, \mathsf{DDHvf}) \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$ and for each $(g, g^x, g^y, g^z) \in \mathbb{G}^4$,

$$\Pr[\mathsf{DDHvf}(g, g^x, g^y, g^z) = (xy = z)] \geq 1 - neg(k)$$

where the probability is taken over all internal coin tosses of DDHvf and "$xy = z$" is defined as 1 is $xy = z \bmod p$ and 0 otherwise. We use $\mathbb{G}^*$ to denote $\mathbb{G} \setminus \{0\}$, i.e. the set of all group elements except the neutral element. Throughout the paper we use $\mathcal{GG} = (\mathbb{G}, p, \mathsf{DDHvf})$ as shorthand for the description of the gap group. See [25] for a more formal treatment of gap groups. We note that one *specific instantiation* of such gap-groups can be obtained using bilinear pairings [6].

5.2 The Decision Linear Assumption.

Let $\mathcal{GG}$ as above and let $g_1, g_2, z \in \mathbb{G}$ be random elements from group $\mathbb{G}$. Consider the following problem introduced By Boneh, Boyen, and Shacham [4]: Given $(g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w) \in \mathbb{G}^6$ as input, output yes if $w = z^{r_1+r_2}$ and no otherwise. One can easily show that an algorithm for solving the Decision Linear Problem in $\mathbb{G}$ gives an algorithm for solving DDH in $\mathbb{G}$. The converse is believed to be false. That is, it is believed that the Decision Linear Problem is a hard problem even in gap-groups where DDH is easy. To an adversary $\mathcal{A}$ we associate the following experiment.

Experiment $\mathbf{Exp}^{\mathrm{dlin}}_{\mathcal{G},\mathcal{A}}(1^k)$
$\mathcal{PG} \xleftarrow{\$} \mathcal{G}(1^k)$; $g_1, g_2, z \xleftarrow{\$} \mathbb{G}^*$; $r_1, r_2, r \xleftarrow{\$} \mathbb{Z}_p$
$\beta \xleftarrow{\$} \{0,1\}$; if $\beta = 1$ then $w \leftarrow z^{r_1+r_2}$ else $w \leftarrow z^r$
$\beta' \xleftarrow{\$} \mathcal{A}(1^k, \mathcal{PG}, g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w)$
If $\beta \neq \beta'$ then return 0 else return 1

We define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{\mathrm{dlin}}_{\mathcal{G},\mathcal{A}}(k) \;=\; \left| \Pr\left[\mathbf{Exp}^{\mathrm{dlin}}_{\mathcal{G},\mathcal{B}}(1^k) = 1 \right] - \frac{1}{2} \right| .$$

We say that the *decision linear assumption relative to generator* $\mathcal{G}$ holds if $\mathbf{Adv}^{\mathrm{dlin}}_{\mathcal{G},\mathcal{A}}$ is a negligible function in k for all PTAs $\mathcal{A}$.

To put more confidence in the DLIN problem it was shown in [4] that the DLIN problem is hard in generic gap-groups.

A BASIC SCHEME BASED ON DLIN. Since it's introduction the DLIN assumption has already found some interesting applications (e.g., see [4, 8, 24]). As noted in [4] the DLIN assumption readily gives a CPA-secure PKE scheme (called linear encryption scheme) as follows: The public key consists of random elements $g_1, g_2, z \in \mathbb{G}$, the secret key of elements x_1, x_2 such that $g_1^{x_1} = g_2^{x_2} = z$. Encryption of a message M is given by $(C_1, C_2, E) \leftarrow (g_1^{r_1}, g_2^{r_2}, z^{r_1+r_2} \cdot M)$, where $r_1, r_2 \in \mathbb{Z}_q^*$ are random elements. The message M is recovered by the possessor of the secret key by computing M as $M \leftarrow E/(C_1^{x_1} C_2^{x_2})$.

5.3 The Scheme

The starting point of our scheme will be the (CPA-secure) linear encryption scheme from Section 5.2. By adding two additional values to the ciphertext we can update it to a selective-tag CCA-secure TBE scheme. The values contain redundant information and also depend on the tag. In the decryption algorithm the two values are used to check the ciphertext for "validity" or "consistency". We build a TBE scheme $\mathcal{TBE} = (\mathsf{TBEkg}, \mathsf{TBEenc}, \mathsf{TBEdec})$ as follows:

DLIN-based TBE

$\mathsf{TBEkg}(1^k)$
- $(\mathbb{G}, p, \mathsf{DDHvf}) \xleftarrow{\$} \mathcal{G}(1^k)$
- $g_1 \xleftarrow{\$} \mathbb{G}^*$; $x_1, x_2, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p^*$
- Chose $g_2, z \in \mathbb{G}$ with $g_1^{x_1} = g_2^{x_2} = z$
- $u_1 \leftarrow g_1^{y_1}$; $u_2 \leftarrow g_2^{y_2}$
- $pk \leftarrow (\mathbb{G}, p, g_1, g_2, z, u_1, u_2)$; $sk \leftarrow (x_1, x_2, y_1, y_2)$
- Return (pk, sk)

$\mathsf{TBEenc}(pk, t, M)$
- $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$
- $C_1 \leftarrow g_1^{r_1}$; $C_2 \leftarrow g_2^{r_2}$
- $D_1 \leftarrow z^{tr_1} u_1^{r_1}$; $D_2 \leftarrow z^{tr_2} u_2^{r_2}$
- $K \leftarrow z^{r_1+r_2}$
- $E \leftarrow M \cdot K$
- $C_{tbe} \leftarrow (C_1, C_2, D_1, D_2, E)$
- Return C_{tbe}

$\mathsf{TBEdec}(sk, t, C_{tbe})$
- Parse C_{tbe} as (C_1, C_2, D_1, D_2, E)
- $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p^*$
- $K \leftarrow \frac{C_1^{x_1+s_1(tx_1+y_1)} \cdot C_2^{x_2+s_2(tx_2+y_2)}}{D_1^{s_1} \cdot D_2^{s_2}}$
- $M \leftarrow E \cdot K^{-1}$
- Return M

Note that the public key pk does not contain the description of the Diffie-Hellman verification oracle DDHvf.

5.4 Correctness and Alternative Decryption

Let $C_{tbe} = (C_1, C_2, D_1, D_2, E) \in \mathbb{G}^5$ be a (possibly malformed) ciphertext. C_{tbe} is called *consistent with tag* t if $C_1^{tx_1+y_1} = D_1$ and $C_2^{tx_2+y_2} = D_2$. Note that any ciphertext that was properly generated by the encryption algorithm for tag t is always consistent with (the same) tag t, i.e. for $i = 1, 2$ we have $(g_i^{r_i})^{tx_i+y_i} = z^{tr_i} u_i^{r_i}$ for any $r_i \in \mathbb{Z}_p$.
The key K in the decryption algorithm is computed as

$$K = \frac{C_1^{x_1+s_1(tx_1+y_1)} C_2^{x_2+s_2(tx_2+y_2)}}{D_1^{s_1} D_2^{s_2}} = C_1^{x_1} C_2^{x_2} \cdot \left(\frac{C_1^{tx_1+y_1}}{D_1}\right)^{s_1} \cdot \left(\frac{C_2^{tx_2+y_2}}{D_2}\right)^{s_2}$$

for uniform $s_1, s_2 \in \mathbb{Z}_q$. This can be viewed as an implicit test if the ciphertext is consistent with tag t. If so the key is computed as $K = C_1^{x_1} \cdot C_2^{x_2}$. If not then at least one of the two fractions in the above equation is different from $1 \in \mathbb{G}$ and (since $\mathbb{G}$ has prime order) a random key K is returned, completely independent of the "real key" $C_1^{x_1} \cdot C_2^{x_2}$. Hence the decryption algorithm in the above construction is equivalent to the following (less efficient) decryption algorithm:

$\mathsf{TBEdec}'(sk, t, C_{tbe})$
- Parse C_{tbe} as (C_1, C_2, D_1, D_2, E)
- If $C_1^{tx_1+y_1} \neq D_1$ or $C_2^{tx_2+y_2} \neq D_2$ then $K \xleftarrow{\$} \mathbb{G}^*$
- Else $K \leftarrow C_1^{x_1} \cdot C_2^{x_2}$
- Return $M \leftarrow E \cdot K^{-1}$

It leaves to verify that, in case the ciphertext is consistent, $K \leftarrow C_1^{x_1} \cdot C_2^{x_2}$ computes the correct key. Indeed we have $(g_1^{r_1})^{x_1} \cdot (g_2^{r_2})^{x_2} = z^{r_1} \cdot z^{r_2} = z^{r_1+r_2}$. This shows correctness.

5.5 Public Verification

In this section we show that consistency (or validity) of a given TBE ciphertext can be publicly verified. The above alternative decryption procedure TBEdec' gives rise to an algorithm $\mathsf{TBEpv}(pk, t, C_{tbe})$ for public verification of the ciphertext by checking if $(g_1, z^t u_1, C_1, D_1)$ and $(g_2, z^t u_2, C_2, D_2)$ are Diffie-Hellman tuples. Both checks can be carried out using the Diffie-Hellman verification algorithm DDHvf that we additionally have to provide in the public-key. To verify correctness of the above public consistency check we have to show that for $i = 1, 2$, $C_i^{tx_i+y_i} = D_i$ iff $(g_i, z^t u_i, C_i, D_i)$ is a Diffie-Hellman tuple. Let $C_i = g^{r_i}$. Then $(g_i, z^t u_i = g_i^{x_i t + y_i}, C_i = g_i^{r_i}, D_i)$ is a proper Diffie-Hellman-tuple iff $g_i^{(x_i t+y_i)\cdot r_i} = D_i$ iff $C_i^{x_i t+y_i} = D_i$.

5.6 Security and Efficiency

Theorem 2. *Under the decision linear assumption relative to generator $\mathcal{G}$, the TBE scheme from Section 5.3 is selective-tag secure against chosen-ciphertext attacks.*

Theorem 2 is proved in Appendix A. The intuition of the proof is as follows: Given an adversary $\mathcal{A}$ against the security of the TBE scheme, we can build an adversary $\mathcal{B}$ that breaks the linear assumption with the same success probability of $\mathcal{A}$. For simulating $\mathcal{A}$'s view we use two main ingredients: First, when answering the decryption queries, $\mathcal{B}$ can test for consistency using the public ciphertext verification algorithm TBEpv from Section 5.5. (This is the reason why pairings are needed for the security proof.) Second, we borrow techniques from [3] to make sure that $\mathcal{B}$ can answer the (consistent) decryption queries for all tags but for the target tag t^* output by $\mathcal{A}$ in the beginning of the security experiment.

Encryption requires three exponentiations (to compute C_1, C_2 and K) and two multi-exponentiation (to compute D_1, D_2) in $\mathbb{G}$. Encryption may as well be carried out in 7 exponentiations what is considerably faster when the receiver's public key is considered to be fixed and precomputation for fixed-base exponentiation is used. Decryption is very fast and can be done with one multi-exponentiation.

6 Key Encapsulation based on the Linear Assumption

A *key encapsulation mechanism* [29] (KEM) $\mathcal{KEM} = (\mathsf{KEMkg}, \mathsf{KEMencaps}, \mathsf{KEMdecaps})$ consits of three PTAs can be seen as a light PKE scheme. Instead of encrypting messages, the encapsulation algorithm $\mathsf{KEMencaps}$ generates a (random) symmetric key K and a corresponding ciphertext C. The decapsulation algorithm inputs the secret key and a ciphertext and reconstructs the symmetric key K. In practice the key K is usually fed to a symmetric encryption scheme. CCA-security of a KEM can be analogously defined as CCA-security security of a PKE scheme; in the security game an adversary is given a ciphertext/key pair and has to decide if the two pairs match or if the key is random and independent from the ciphertext. A formal definition of a CCA-secure KEM can be looked up in the full version [20].

6.1 The KEM Scheme

We build a KEM scheme as follows. Let $\mathsf{KEMkg}(1^k)$ be as in the TBE scheme of Section 5.3. The public key pk additionally contains a target collision resistant hash function $\mathsf{TCR} : \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q$ (i.e. given $t = \mathsf{TCR}(g_1, g_2)$ it should be hard to find $(h_1, h_2) \in \mathbb{G} \times \mathbb{G} \setminus \{(g_1, g_2)\}$ such that $\mathsf{TCR}(h_1, h_2) = t$; we refer to [12] for a formal definition).[5] The encapsulation/decapsulation algorithms are as follows:

DLIN-based KEM	
$\mathsf{KEMencaps}(pk)$	$\mathsf{KEMdecaps}(sk, C_{kem})$
$r_1, r_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$	Parse C_{kem} as (C_1, C_2, D_1, D_2)
$C_1 \leftarrow g_1^{r_1}$; $C_2 \leftarrow g_2^{r_2}$	$t \leftarrow \mathsf{TCR}(C_1, C_2)$
$t \leftarrow \mathsf{TCR}(C_1, C_2)$	$s_1, s_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$
$D_1 \leftarrow z^{tr_1} u_1^{r_1}$; $D_2 \leftarrow z^{tr_2} u_2^{r_2}$	$K \leftarrow \frac{C_1^{x_1+s_1(tx_1+y_1)} \cdot C_2^{x_2+s_2(tx_2+y_2)}}{D_1^{s_1} \cdot D_2^{s_2}}$
$K \leftarrow z^{r_1+r_2}$	Return K
$C_{kem} \leftarrow (C_1, C_2, D_1, D_2)$	
Return (C_{kem}, K)	

Analogous to the TBE construction from Section 5 consistency of a ciphertext $C_{kem} = (C_1, C_2, D_1, D_2)$ can be publicly verified by computing $t \leftarrow \mathsf{TCR}(C_1, C_2)$ and checking if $(g_i, z^t u_i, C_i, D_i)$ is a Diffie-Hellman tuple for $i = 1, 2$.

6.2 Security

Theorem 3. *Assume* TCR *is a target collision resistant hash function. Under the decision linear assumption relative to the generator* $\mathcal{G}$ *the KEM from Section 6.1 is secure against chosen-ciphertext attacks.*

The security reduction is tight and compared to the reduction from Theorem 2 there appears an additional additive factor taking into account a possible collision in the hash function TCR. The proof of Theorem 3 is similar to that of Theorem 2 and is given in the full version [20].

The way we use the target collision hash function is reminiscent to the Cramer-Shoup cryptosystem [12]. Indeed, the intuition is the same. Given an adversary $\mathcal{A}$ against the security of the KEM, we can build an adversary $\mathcal{B}$ that breaks the linear assumption with the same success probability of $\mathcal{A}$. Let $(C_1^*, C_2^*, D_1^*, D_2^*)$ be the challenge ciphertext given to adversary $\mathcal{A}$ and let $t^* = \mathsf{TCR}(C_1^*, C_2^*)$. Consider a ciphertext (C_1, C_2, D_1, D_2) queried by adversary $\mathcal{A}$ during the CCA experiment and let $t = \mathsf{TCR}(C_1, C_2)$. Similar to the proof of Theorem 2 we can setup the public-key in a way such that $\mathcal{B}$ is able to correctly

[5] More formally we need a family of hash functions indexed by some random key c, where c is contained in the public key and the description of the hash function is included in the scheme parameters.

simulate all such decryption queries as long as $t \neq t^*$ and the ciphertext is constentent. The latter one can be checked using the public consistency algorithm. Assume $t = t^*$. On one hand, when $(C_1, C_2) \neq (C_1^*, C_2^*)$ then $\mathcal{B}$ found a collision in the hash function. On the other hand, when $(C_1, C_2) = (C_1^*, C_2^*)$ then consistency of the ciphertext also implies $D_1 = D_1^*$ and $D_2 = D_2^*$ and hence the queried ciphertext matches the target ciphertext what is forbidden in the experiment.

6.3 From KEM to Full PKE

It is well known that if both the public-key encapsulation scheme and the underlying symmetric-key encryption scheme are CCA-secure, then the resulting hybrid public-key encryption scheme is CCA-secure [13, Sec. 7]. The security reduction is tight.

7 Discussion

7.1 Efficiency considerations

An efficiency comparison of all previously known CCA-secure PKE schemes in the standard model is assembled in Figure 2. The Cramer-Shoup scheme [12] and the Kurosawa-Desmedt scheme [21] are listed for reference. BK/BBx refers to one of the two Boneh-Boyen IBE schemes from [3] instantiated with the MAC based BK-transformation (since the signature-based CHK transformation is less efficient we decided not to list it in our comparison).

BMW is the recent KEM from Boyen, Mei, and Waters [9]. To obtain a fair comparison we equipped the two KEM schemes (the BMW-KEM and ours from §6) with a hybrid encryption scheme to obtain a fully fledged PKE scheme.

Together with the Kurosawa-Desmedt PKE, our proposed DLIN-based KEM offers the nowadays fastest decryption algorithm. Compared to all other schemes the obvious drawbacks of our schemes are slower encryption and longer ciphertexts. Interestingly, the BK/BBx and BMW constructions tie with the KD scheme in terms of encryption but lose in terms of decryption, whereas our scheme loses in encryption but ties in decryption.

We note that the long ciphertexts are basically due to the different assumption; this is since the basic (chosen-plaintext secure) linear encryption scheme from Section 5.2 already comes with a ciphertext overhead of $2|p|$.

7.2 Remarks

We hope that by having provided weaker sufficient conditions for the CHK/BK transformations we make a step directed towards a better understanding and utilization of CCA-security in PKE schemes. From a designer's point of view the definition of selective-tag security means that the scheme only has to be "secured" with respect to the target tag. Furthermore, in the security reduction, the

Scheme	Origin	Assumption	Encryption	Decryption	Ciphertext Overhead	Public Vfy?
			#pairings + #[multi,reg,fix]-exp			
KD	direct	DDH	$0 + [1, 2, 0]$	$0 + [1, 0, 0]$	$2\|p\|$(+hybrid)	—
CS	KEM	DDH	$0 + [1, 3, 0]$	$0 + [1, 1, 0]$	$3\|p\|$	—
BK/BB1	BK/IBE	BDDH	$0 + [1, 2, 0]$	$1 + [1, 0, 0]$	$2\|p\|$+com+mac	—
BK/BB2	BK/IBE	q-BDDHI	$0 + [1, 2, 0]$	$1 + [0, 1, 1]$	$2\|p\|$+com+mac	—
BMW	KEM	BDDH	$0 + [1, 2, 0]$	$1 + [0, 1, 0]$	$2\|p\|$	yes
Ours (§5)	BK/TBE	DLIN	$0 + [2, 3, 0]$	$0 + [1, 0, 0]$	$4\|p\|$+com+mac	—
Ours (§6)	KEM	DLIN	$0 + [2, 3, 0]$	$0 + [1, 0, 0]$	$4\|p\|$	yes

Fig. 2. Efficiency comparison for CCA-secure PKE schemes. Some figures are borrowed from [7, 5, 9]. All "private-key" operations (such as hash function/MAC/KDF) are ignored. Cipher overhead represents the difference (in bits) between the ciphertext length and the message length, and $|p|$ is the length of a group element. For concreteness one can think of mac = 128 and the commitment com = 512 bits. For comparison we mention that relative timings for the various operations are as follows: bilinear pairing ≈ 5 [28], multi-exponentiation ≈ 1.5, regular exponentiation $= 1$, fixed-base exponentiation $\ll 0.2$.

generated keys may depend on this tag. Having that designing concept in mind it would be interesting to come up with new CCA-secure TBE/PKE schemes based on different assumptions.

A very efficient TBE construction based on the Kurosawa-Desmedt encryption scheme [21] is obtained by removing the target collission-resistant hash function and taking the former output of the hash function as the tag. A straightforward question is if we can somewhat modify either this KD based TBE scheme or our proposal from Section 5 to obtain an IBE scheme that does not use pairings.

7.3 Acknowledgments

We thank Mihir Bellare, Xavier Boyen, Yoshi Kohno, Gregory Neven, and the anonymous TCC referees for useful remarks.

References

1. M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 128–146. Springer-Verlag, May 2005.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.
3. D. Boneh and X. Boyen. Efficient selective-id secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, May 2004.
4. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, Aug. 2004.

5. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. Journal submission. Available from author's web page `http://crypto.stanford.edu/~dabo/pubs.html`, November 2005.
6. D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
7. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer-Verlag, Feb. 2005.
8. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *ACM CCS 04*, pages 168–177. ACM Press, Oct. 2004.
9. X. Boyen, Q. Mei, and B. Waters. Simple and efficient CCA2 security from IBE techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*, pages 320–329. New-York: ACM Press, 2005.
10. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.
11. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, May 2004.
12. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, Aug. 1998.
13. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
14. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1978.
15. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
16. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer-Verlag, Aug. 1985.
17. E. Elkind and A. Sahai. 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. D. Galindo and I. Hasuo. Security notions for identity based encryption. Cryptology ePrint Archive, Report 2005/253, 2005. `http://eprint.iacr.org/`.
19. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
20. E. Kiltz. Chosen-ciphertext security from tag-based encryption. Cryptology ePrint Archive, 2005. `http://eprint.iacr.org/`.
21. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer-Verlag, Aug. 2004.
22. P. D. MacKenzie, M. K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions, and applications. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 171–190. Springer-Verlag, Feb. 2004.
23. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*. ACM Press, May 1990.
24. L. Nguyen and R. Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 372–386. Springer-Verlag, Dec. 2004.

25. T. Okamoto and D. Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In K. Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118. Springer-Verlag, Feb. 2001.
26. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer-Verlag, Aug. 1991.
27. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
28. M. Scott. Faster pairings using an elliptic curve with an efficient endomorphism. Cryptology ePrint Archive, Report 2005/252, 2005. `http://eprint.iacr.org/`.
29. V. Shoup. A proposal for an ISO standard for public key encryption (version 2.1). manuscript, 2001. Available on `http://shoup.net/papers/`.
30. B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, May 2005.

A Proof of Theorem 2

Adversary $\mathcal{B}$ inputs an instance of the decisional linear problem, i.e. $\mathcal{B}$ inputs the values $(1^k, \mathcal{GG}, g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w)$. $\mathcal{B}$'s goal is to determine whether $w = z^{r_1+r_2}$ or w is a random group element.

Now suppose there exists an adversary $\mathcal{A}$ that breaks the selective-tag CCA security of the TBE scheme with (non-negligible) advantage $\mathbf{Adv}^{\text{tbe-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)$. We show that adversary $\mathcal{B}$ can run adversary $\mathcal{A}$ to solve its instance of the decisional linear problem (i.e. to determine whether $w = z^{r_1+r_2}$ or if w is a random group element) with advantage

$$\mathbf{Adv}^{\text{dlin}}_{\mathcal{G},\mathcal{B}}(k) \geq \mathbf{Adv}^{\text{tbe-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k) . \tag{1}$$

Now Eqn. (1) proves the Theorem. Adversary $\mathcal{B}$ runs adversary $\mathcal{A}$ simulating its view as in the original TBE security experiment. We now give the description of adversary $\mathcal{B}$.

Init Stage Adversary $\mathcal{B}$ runs adversary $\mathcal{A}$ on input 1^k and `init`. $\mathcal{A}$ outputs the target tag t^* that is input by $\mathcal{B}$.

Find Stage $\mathcal{B}$ picks two random values $c_1, c_2 \in \mathbb{Z}_p$ and sets

$$u_1 \leftarrow z^{-t^*} \cdot g_1{}^{c_1}, \qquad u_2 \leftarrow z^{-t^*} \cdot g_2{}^{c_2} .$$

The public key pk is defined as $(\mathbb{G}, p, g_1, g_2, z, u_1, u_2)$ and it is identically distributed as in the original TBE scheme. Let $x_1 = \log_{g_1} z$ and $x_2 = \log_{g_2} z$, as in the original TBE scheme. This implicitly defines the values y_1, y_2 as

$$y_1 = \log_{g_1} u_1 = -t^* x_1 + c_1, \qquad y_2 = \log_{g_2} u_2 = -t^* x_2 + c_2 .$$

Note that no value of the corresponding secret key $sk = (x_1, x_2, y_1, y_2)$ is known to $\mathcal{B}$.

Now consider an arbitrary ciphertext $C_{tbe} = (C_1, C_2, D_1, D_2)$ and let $t \in \mathbb{Z}_p$ be a tag. Recall that C_{tbe} is consistent with tag t if $C_i^{x_i \cdot t + y_i} = D_i$ for $i \in \{1, 2\}$. The way the keys are setup this condition can be rewritten as

$$D_i = C_i^{tx_i + y_i} = C_i^{x_i t - t^* x_i + c_i} = (C_i^{x_i})^{t-t^*} \cdot C_i^{c_i}, \quad i \in \{1, 2\} . \tag{2}$$

By Equation (2), $D_i / C_i^{c_i} = (C_i^{x_i})^{t-t^*}$ and if $t \neq t^*$ then the session key $K = C_1^{x_1} \cdot C_2^{x_2}$ can alternatively be reconstructed as

$$K \leftarrow \left(\frac{D_1 \cdot D_2}{C_1^{c_1} \cdot C_2^{c_2}} \right)^{\frac{1}{t-t^*}} . \tag{3}$$

Now adversary $\mathcal{B}$ runs $\mathcal{A}$ on input `find` and pk answering to its decryption queries as follows: Let $C_{tbe} = (C_1, C_2, D_1, D_2)$ be an arbitrary ciphertext submitted to the decryption oracle $\textsc{Dec}(C_{tbe}, t)$ for tag $t \neq t^*$. First $\mathcal{B}$ performs a public consistency check as explained in Section 5.5 using the Diffie-Hellman verification algorithm DDHvf. If C_{tbe} is not consistent then $\mathcal{B}$ returns a random message, as in the alternative (but equivalent) decryption algorithm (Section 5.4) of the original TBE scheme. Otherwise, if the ciphertext is consistent adversary $\mathcal{B}$ computes the session key by Equation (3) as $K \leftarrow (\frac{D_1 D_2}{C_1^{c_1} C_2^{c_2}})^{\frac{1}{t-t^*}}$ and returns $M \leftarrow E \cdot K^{-1}$. This shows that as long as $t \neq t^*$ the simulation of the decryption queries is always perfect, i.e. the output of oracle $\textsc{Dec}(C_{tbe}, t)$ is identically distributed as the output of $\mathsf{TBEdec}(sk, C_{tbe}, t)$.

Guess Stage $\mathcal{A}$ returns two distinct messages M_0, M_1 of equal length. Adversary $\mathcal{B}$ picks a random bit b and constructs the challenge ciphertext $C_{tbe}^* = (C_1^*, C_2^*, D_1^*, D_2^*, E^*)$ for message M_b as follows:

$$(\ C_1^* = g_1^{r_1},\ C_2^* = g_2^{r_2},\ D_1^* = (g_1^{r_1})^{c_1},\ D_2^* = (g_2^{r_2})^{c_2},\ E^* = M_b \cdot w\ )$$

By Equation (2), C_{tbe}^* is always consistent with target tag t^*. If $w = z^{r_1 + r_2}$, then $E = M_b \cdot w$ is indeed a valid ciphertext of message M_b and tag t^* under the public key pk. On the other hand, when w is uniform and independent in $\mathbb{G}$ then $E = w \cdot M_b$ is independent of b in the adversary's view.

Adversary $\mathcal{A}$ is run with challenge ciphertext C_{tbe}^* answering to its decryption queries as in the find stage.

Eventually, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows: If $b = b'$ then $\mathcal{B}$ outputs 1 meaning $w = z^{r_1 + r_2}$. Otherwise, it outputs 0 meaning that w is random.

This completes the description of adversary $\mathcal{B}$. We now analyze $\mathcal{B}$'s success in breaking the decisional linear problem.

When the value w input by $\mathcal{B}$ equals to $w = z^{r_1 + r_2}$, then $\mathcal{A}$'s view is identical to its view in a real attack game and therefore $\mathcal{A}$ must satisfy $|\Pr[b = b'] - 1/2| \geq \mathbf{Adv}_{\mathcal{TBE},\mathcal{A}}^{\text{tbe-stag-cca}}(k)$. On the other hand, when w is uniform in $\mathbb{G}$ then $\Pr[b = b'] = 1/2$. Therefore $\mathbf{Adv}_{\mathcal{G},\mathcal{B}}^{\text{dlin}}(k) \geq \left| \left(\frac{1}{2} \pm \mathbf{Adv}_{\mathcal{TBE},\mathcal{A}}^{\text{tbe-stag-cca}}(k) \right) - \frac{1}{2} \right| = \mathbf{Adv}_{\mathcal{TBE},\mathcal{A}}^{\text{tbe-stag-cca}}(k)$. This proves Equation (1) and concludes the proof.