A preliminary version of this paper appears in the proceedings of the third Theory of Cryptography Conference, TCC 2006, Lecture Notes in Computer Science Vol. 3876, Shai Halevi and Tal Rabin eds, Springer Verlag, 2006. This is the full version.

Chosen-Ciphertext Security from Tag-Based Encryption

Eike Kiltz

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 Gap Decisional Linear Assumption. 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.

Keywords: Foundations, chosen-ciphertext security, KEM

1 Introduction

Since Diffie and Hellman proposed the idea of public key cryptography [15], one of the most active area of research in the field has been the design and analysis of public key encryption (PKE) schemes. In [17, 34] efficient primitives were suggested from which to build encryption schemes. Formal models of security were developed in [21, 29, 33] and nowadays it is widely accepted that security against chosen-ciphertext attacks provides the "right level of security" for public-key encryption schemes.

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., [11]).

Dolev, Dwork, and Naor [16] 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 [13] presented the first really practical public-key encryption scheme. Their approach was further generalized in [14] and later shown by Elkind and Sahai [18] 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 [25, 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 [12] 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 [8] 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 [27] introduce the notion of tag-based encryption (TBE) and show (independent from [12]) 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 [12] 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.

1.1 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 [27] and the IBE definition from [12]. 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

[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 (in contrast to the Decisional Diffie-Hellman assumption for the CS/KD scheme). We note that q-strong BDDHI is a stronger assumption than BDDH.

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 [38] (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 Pairings. To underline the usefulness of our TBE to PKE transformation we present an efficient TBE scheme that (in contrast to all currently known IBE schemes) does not 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 [5] 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 gap DLIN assumption which is the assumption that the DLIN problem is hard relative to a DDH oracle. In *gap-groups* [31], 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 (e.g., when an efficiently computable bilinear pairing is available), this assumption is equivalent to the standard DLIN assumption. The security reduction is tight, simple, and very intuitive.

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 Gap DLIN assumption. We remark that this is the first (practical) chosen-ciphertext secure PKE based on the Gap DLIN assumption in the standard model.[2]

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

[2]The scheme contained in the preliminary version of this paper [24] relied on pairings for its security proof. Here we shat that the Gap DLIN assumption is sufficient.

scheme using symmetric encryption (with appropriate security properties).

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 [13, 25]) 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 Gap DLIN assumption in the standard model.

We also propose a direct KEM construction whose chosen-ciphertext security is tightly related to the Bilinear Decisional Diffie-Hellman (BDDH) assumption in pairing groups. This 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). Compared the our DLIN based KEM it is, is slightly more efficient in terms of encryption operations and comes with smaller ciphertexts. In Section 7 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 [25] can still be considered as the most efficient but our schemes allow for public verification of the ciphertexts.

1.2 Related Work

Independent of our work, Boyen, Mei, and Waters [10] recently look at some specific PKE schemes obtained from the CHK transformation instantiated with the IBE schemes from [3, 39] and show how to make the resulting schemes more efficient (in terms of computation time and ciphertext length). In particular, their work also contains our BDDH-based KEM from Section 6.1.

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 $s \xleftarrow{\$} S$ 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 $z \xleftarrow{\$} \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ 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 $z \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ 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

3.1 Public-Key Encryption

An *public-key encryption* (PKE) scheme $\mathcal{PKE} = (\mathsf{PKE.kg}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ consists of three polynomial time algorithms (PTAs). Via $(pk, sk) \xleftarrow{\$} \mathsf{PKE.kg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $C \xleftarrow{\$} \mathsf{PKE.Enc}(pk, M)$ a sender encrypts a message M under the public key pk to get a ciphertext; via $M \leftarrow \mathsf{PKE.Dec}(sk, C)$ 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 $M \in MsgSp(k)$ we have $\Pr[\mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, M)) = M] = 1$, where the probability is taken over the coins of all the algorithms in the expression above.

PRIVACY. Privacy follows [33]. Let $\mathcal{PKE} = (\mathsf{PKE.kg}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ be an PKE scheme with associated message space $MsgSp$. To an adversary $\mathcal{A}$ we associate the following experiment:

$$\begin{array}{l}\textbf{Experiment } \mathbf{Exp}^{pke\text{-}\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}(k)\\ \quad (pk, sk) \xleftarrow{\$} \mathsf{PKE.kg}(1^k)\\ \quad (M_0, M_1, St) \xleftarrow{\$} \mathcal{A}^{\mathrm{DEC}(\cdot)}(\mathtt{find}, pk)\\ \quad b \xleftarrow{\$} \{0,1\}\ ;\ C^* \xleftarrow{\$} \mathsf{PKE.Enc}(pk, M_b)\\ \quad b \xleftarrow{\$} \mathcal{A}^{\mathrm{DEC}(\cdot)}(\mathtt{guess}, C^*, St)\\ \quad \text{If } b \neq b \text{ then return 0 else return 1}\end{array}$$

where the oracle $\mathrm{DEC}(C)$ returns $M \leftarrow \mathsf{PKE.Dec}(sk, C)$ with the restriction that in the `guess` phase adversary $\mathcal{A}$ is not allowed to query oracle $\mathrm{DEC}(\cdot)$ 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 $MsgSp(k)$. We define the advantage of $\mathcal{A}$ in the above experiment as

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

PKE scheme $\mathcal{PKE}$ is said to be *secure against chosen ciphertext attacks* (CCA-secure) if the advantage function $\mathbf{Adv}^{pke\text{-}\mathrm{cca}}_{\mathcal{PKE},\mathcal{A}}$ 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 [27], 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 [12] and was not considered in [27].

More formally, a *tag-based encryption* (TBE) scheme $\mathcal{TBE} = (\mathsf{TBE.kg}, \mathsf{TBE.Enc}, \mathsf{TBE.Dec})$ consists of three PTAs. Via $(pk, sk) \xleftarrow{\$} \mathsf{TBE.kg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $C \xleftarrow{\$} \mathsf{TBE.Enc}(pk, t, M)$ a sender encrypts a message M with tag t to get a ciphertext; via $M \xleftarrow{\$} \mathsf{TBE.Dec}(sk, t, C)$ 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. We also stress that $\mathsf{TBE.Dec}$ may be probabilistic. 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 $M \in MsgSp(k)$ we have $\Pr[\mathsf{TBE.Dec}(sk, t, \mathsf{TBE.Enc}(pk, t, M)) = M] = 1$, where the probability is taken over the choice of $(pk, sk) \xleftarrow{\$} \mathsf{TBE.kg}(1^k)$, and the coins of all the algorithms in the expression above.

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

$$\textbf{Experiment } \mathbf{Exp}^{\textit{tbe}\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)$$
$(t^*, St_0) \xleftarrow{\$} \mathcal{A}(1^k, \texttt{init})$
$(pk, sk) \xleftarrow{\$} \mathsf{TBE.kg}(1^k)$
$(M_0, M_1, St) \xleftarrow{\$} \mathcal{A}^{\textsc{Dec}(\cdot,\cdot)}(\texttt{find}, pk, St_0)$
$b \xleftarrow{\$} \{0,1\}\ ;\ C^* \xleftarrow{\$} \mathsf{TBE.Enc}(pk, t^*, M_b)$
$b \xleftarrow{\$} \mathcal{A}^{\textsc{Dec}(\cdot,\cdot)}(\texttt{guess}, C^*, St)$
If $b \neq b$ then return 0 else return 1

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

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

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^* (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 [38] (and were called "labels") and later by MacKenzie, Reiter, and Yang [27]. 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 [38, 27] and point out the differences to our definition. Let C^* 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 [27][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 [38]), we further modify the security experiment (of weak CCA security) such that the adversary is allowed to ask any decryption query suspect to $(t, C) \neq (t^*, C^*)$. In particular this includes queries for the target tag t^* as long as $C \neq C^*$.

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

[3]Note that weak CCA-security for TBE schemes was called CCA-security in [27]. But for its relation to PKE schemes we prefer to refer to it as weak CCA-security. This should become clear later.

TBE security	Restriction to $\textsc{Dec}(t, C)$ queries	Selective-tag?
(full) CCA [38]	$(t, C) \neq (t^*, C^*)$	no
weak CCA [27]	$t \neq t^*$	no
selective-tag weak CCA (ours)	$t \neq t^*$	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 [19] 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^*, 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. Figure 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, the key derivation algorithm $\mathsf{IBE.Keyder}$. On input of the secret key sk and an identity id, $\mathsf{IBE.Keyder}$ 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 Appendix A.2.

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 Appendix A.2for 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

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

TBE schemes in the weak selective-tag model are easier to construct. Figure 1 in Section 4 is summarizing the relations between TBE and IBE.

3.4 One-Time Signatures

A public key signature scheme $\mathcal{OTS} = (\mathsf{S.Kg}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ with associated message space MsgSp consists of three PTAs. Via $(\mathit{verk}, \mathit{sk}) \xleftarrow{\$} (\mathsf{S.Kg}(1^k)$ the randomized key-generation algorithm produces a key-pair for security parameter $k \in \mathbb{N}$; via $\sigma \xleftarrow{\$} \mathsf{S.Sign}(\mathit{sk}, M)$ the user signs a message $M \in \mathit{MsgSp}(k)$ with his secret key sk to get a signature σ; via $\mathsf{S.Vfy}(\mathit{verk}, M, \sigma)$ the signature σ on the message M is verified with respect to the verification key verk. $\mathsf{S.Vfy}$ outputs OK or $\perp$. For consistency we requite that for all $k \in \mathbb{N}$ and all messages $M \in \mathit{MsgSp}(k)$, we have $\Pr[\mathsf{S.Vfy}(\mathit{verk}, M, \mathsf{S.Sign}(\mathit{sk}, M)) = \mathrm{OK}] = 1$, where the probability is taken over the choice of $(\mathit{verk}, \mathit{sk}) \xleftarrow{\$} (\mathsf{S.Kg}(1^k)$, and the coins of all the algorithms in the expression above.

Security. Let $\mathcal{OTS}$ be a signature scheme, let k be a security parameter, and let $\mathcal{A}$ be an adversary. We consider the following experiment:

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\mathit{ots}\text{-ex-for}}_{\mathcal{OTS},\mathcal{A}}(k) \\
\quad (\mathit{verk}, \mathit{sk}) \xleftarrow{\$} \mathsf{S.Kg}(1^k) \\
\quad (M^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\textsc{Sign}}(\mathtt{find}, \mathit{verk}) \\
\quad \text{If } \mathsf{S.Vfy}(\mathit{sk}, M^*, \sigma^*) = \perp \text{ then return } 0 \\
\quad \text{Return } 1
\end{array}
$$

where the oracle $\textsc{Sign}(M)$ is returns $\sigma \xleftarrow{\$} \mathsf{S.Sign}(\mathit{sk}, M)$ and $\mathcal{A}$ may only make one single query to oracle $\textsc{Sign}(\cdot)$. The target pair (M^*, σ^*) output by adversary $\mathcal{A}$ must be different from the message/signature pair (M, σ) obtained from the oracle query. We define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{\mathit{ots}\text{-ex-for}}_{\mathcal{OTS},\mathcal{A}}(k) \;=\; \Pr\left[\mathbf{Exp}^{\mathit{ots}\text{-ex-for}}_{\mathcal{OTS},\mathcal{A}}(k) = 1\right].$$

Signature scheme $\mathcal{OTS}$ is said to be a *strong one-time signature scheme* if the advantage function $\mathbf{Adv}^{\mathit{ots}\text{-ex-for}}_{\mathcal{OTS},\mathcal{A}}$ is a negligible function in k for all PTAs $\mathcal{A}$.

4 Chosen-Ciphertext Security from Tag-Based Encryption

Canetti, Halevi, and Katz [12] 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 [12], MacKenzie, Reiter, and Yang [27] exploit the same construction as [12] 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 [12, 27, 8] 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 CCA-secure.

4.1 The Transformation

Given a TBE scheme $\mathcal{TBE} = (\mathsf{TBE.kg}, \mathsf{TBE.Enc}, \mathsf{TBE.Dec})$ with tag-space TagSp we construct a public-key encryption scheme $\mathcal{PKE} = (\mathsf{PKE.kg}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$. In the construction, we use

a one-time signature scheme $\mathcal{OTS} = (\mathsf{S.Kg}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ in which the verification key output by $\mathsf{S.Kg}(1^k)$ is an element from $TagSp$. (If that is not the case we can apply a target colission resistant hash function that maps the verification keys to $TagSp$.) We require that this scheme be secure in the sense of strong unforgeability. 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. $\mathsf{PKE.kg}(1^k)$ outputs whatever $\mathsf{TBE.kg}(1^k)$ outputs. The construction proceeds as follows:

$\mathsf{PKE.Enc}(pk, M)$	$\mathsf{PKE.Dec}(sk, C)$
$(verk, sigk) \xleftarrow{\$} \mathsf{S.Kg}(1^k)$	Parse C as $(C, verk, \sigma)$
$C \xleftarrow{\$} \mathsf{TBE.Enc}(pk, verk, M)$	If $\mathsf{S.Vfy}(verk, C, \sigma) = \perp$
$\sigma \xleftarrow{\$} \mathsf{S.Sign}(sigk, C)$	then return $\perp$.
Return $C \leftarrow (C, verk, \sigma)$	Else return $M \xleftarrow{\$} \mathsf{TBE.Dec}(sk, verk, C)$

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 $(C^*, verk^*, \sigma^*)$ 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^* is output by $\mathsf{TBE.Enc}$ which is CPA-secure, $verk^*$ is independent of the message, and σ^* is the result of applying the one-time signing algorithm to C^*.

We claim that decryption oracle queries cannot further help the adversary in guessing the value of b. Consider an arbitrary ciphertext query $(C, verk, \sigma) \neq (C^*, verk^*, \sigma^*)$ made by the adversary during the experiment. If $verk = verk^*$ then $(C, \sigma) \neq (C^*, \sigma^*)$ and the decryption oracle will answer $\perp$ since the adversary is unable to forge a new valid signature σ with respect to $verk^*$. If $verk \neq verk^*$ then the decryption query will not help the adversary since the actual decryption using $\mathcal{TBE}$ will be done with respect to a tag $verk$ different to the target tag $verk^*$. A formalization of the above arguments leads to the following:

Theorem 4.1 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 [12, 6] 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.

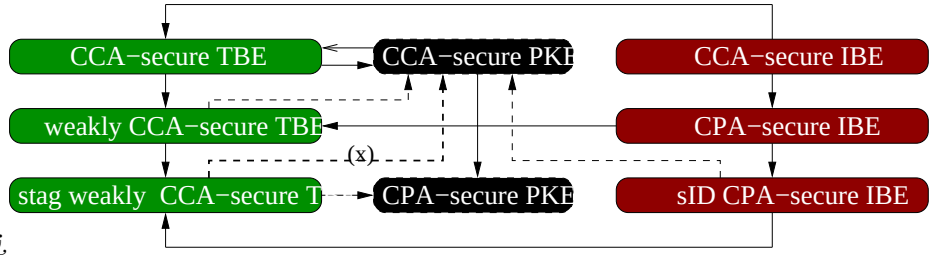
For simplicity we only described the CHK transformation in this Section. We want to remark that the more efficient BK transformation [8, 6] (which basically employs a MAC insteas of a signature) works as well for TBE schemes. The use of a MAC instead of a one-time signature somewhat complicates exposition and proof.

4.2 Classifying the different TBE security notions

Since from a complexity-theoretic point of view strong one-time signatures can be black-box constructed from any one-way function [35, 26, 20] (and hence from any TBE scheme) we can draw the following corollary.

Corollary 4.2 The class of public-key encryption schemes secure against chosen-ciphertext attacks and the class of tag-based encryption schemes selective-tag secure against chosen plaintext attacks are equivalent through a black-box polynomial-time reduction.

M¡relations.pdf, id=51, 480.79625pt x



128.48pt¿M¡use relations.pdf¿

Figure 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 [27], the right one due to [12], and the one with the marker (x) shows our contribution.

This Corollary implies that all three TBE definitions from Section 3.2 are in fact *equivalent* through a polynomial-time black-box reductions. We summarize the known relations among TBE, PKE, and IBE in Figure 1. The results of this section settle the implication marked by (x).

5 Constructions of TBE schemes

5.1 Generic construction from any IBE scheme

Given an IBE scheme $\mathcal{IBE} = (\mathsf{IBE.kg}, \mathsf{IBE.Enc}, \mathsf{IBE.Keyder}, \mathsf{IBE.Dec})$ with identity-space $IDSp$ we construct a TBE scheme $\mathcal{TBE} = (\mathsf{TBE.kg}, \mathsf{TBE.Enc}, \mathsf{TBE.Dec})$ with tag-space $TagSp = IDSp$. The transformation defines the public/secret key pair of the TBE scheme to be the public/secret key pair of the IBE scheme, i.e. $\mathsf{TBE.kg}(1^k)$ outputs whatever $\mathsf{IBE.kg}(1^k)$ outputs. The construction proceeds as follows:

$\mathsf{TBE.Enc}(pk, id, M)$	$\mathsf{TBE.Dec}(sk, id, C)$
$C \xleftarrow{\$} \mathsf{IBE.Enc}(pk, id, M)$	$sk[id] \xleftarrow{\$} \mathsf{IBE.Keyder}(sk, id)$
Return C	$M \leftarrow \mathsf{IBE.Dec}(pk, sk[id], C)$

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

Theorem 5.1 Suppose $\mathcal{IBE}$ is a {CCA, CPA, selective-id CPA}-secure IBE scheme. Then $\mathcal{TBE}$ from the above construction is a {CCA, weakly CCA, selective-tag weakly CPA}-secure TBE scheme.

5.2 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.

5.2.1 The Decision Linear Assumption.

Our scheme will be parameterized by a *parameter generator*. This is a polynomial-time algorithm $\mathcal{G}$ that on input 1^k returns the description of a multiplicative cyclic group $\mathbb{G}$ of prime order p,

where $2^k < p < 2^{k+1}$, a generator g of $\mathbb{G}$. We use $\mathbb{G}^*$ to denote $\mathbb{G} \setminus \{1\}$, i.e. the set of all group elements except the neutral element. Throughout the paper we use $\mathcal{GR} = (\mathbb{G}, g, p)$ as shorthand for the description of the group.

A tuple $(g, g^x, g^y, g^z) \in \mathbb{G}^4$ is called a *Diffie-Hellman tuple* if $xy = z \bmod p$. A DDH 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 $(\mathbb{G}, p, \text{DDHVF}) \xleftarrow{\$} \mathcal{G}(1^k)$ and for each $(g, g^x, g^y, g^z) \in \mathbb{G}^4$,

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

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.

A possible implementation of the DDH oracle is given by the Weil/Tate bilinear pairing allowing to efficiently compute a bilinear pairing which can be used to solve DDH with probability 1 [7].

Let $\mathcal{GR} = (\mathbb{G}, g, p)$ 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 [5]: 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, even with access to a DDH oracle DDHVF. 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 even relative to a DDH oracle. To an adversary $\mathcal{A}$ we associate the following experiment.

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

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

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

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

To put more confidence in the DLIN problem it was shown in [5] that the DLIN problem is hard in bilinear groups.

Since it's introduction the DLIN assumption (in bilinear groups) has already found some interesting applications (e.g., see [5, 9, 30]). As noted in [5] 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. Message M is recovered by the possessor of the secret key by computing $M \leftarrow e/(c_1^{x_1} c_2^{x_2})$.

5.2.2 The Scheme

The starting point of our scheme will be the (CPA-secure) linear encryption scheme from Section 5.2.1. 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 $\mathcal{LTBE} = (\mathsf{LTBE.kg}, \mathsf{LTBE.Enc}, \mathsf{LTBE.Dec})$ as follows:

$\mathsf{LTBE.kg}(1^k)$

$g_1 \xleftarrow{\$} \mathbb{G}^*$; $x_1, x_2, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p$

Chose $g_2, z \in \mathbb{G}$ with $g_1^{x_1} = g_2^{x_2} = z$

$u_1 \leftarrow g_1^{y_1}$; $u_2 \leftarrow g_2^{y_2}$

$pk \leftarrow (g_1, g_2, z, u_1, u_2)$; $sk \leftarrow (x_1, x_2, y_1, y_2)$

Return (pk, sk)

$\mathsf{LTBE.Enc}(pk, t, M)$

$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$

$c_1 \leftarrow g_1^{r_1}$; $c_2 \leftarrow g_2^{r_2}$

$d_1 \leftarrow z^{tr_1} u_1^{r_1}$; $d_2 \leftarrow z^{tr_2} u_2^{r_2}$

$K \leftarrow z^{r_1+r_2}$; $\psi \leftarrow K \cdot M$

$C \leftarrow (c_1, c_2, d_1, d_2, \psi) \in \mathbb{G}^5$

Return C

$\mathsf{LTBE.Dec}(sk, t, C)$

Parse C as $(c_1, c_2, d_1, d_2, \psi)$

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 $M \leftarrow \psi \cdot K^{-1}$

5.2.3 Correctness and Alternative Decryption

Let $C = (c_1, c_2, d_1, d_2, \psi) \in \mathbb{G}^5$ be a (possibly malformed) ciphertext. The KEM part $c = (c_1, c_2, d_1, d_2)$ 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 $r_i \in \mathbb{Z}_p$. In decryption the ciphertext is first checked for consistency. If it is consistent the key is reconstructed as $K = c_1^{x_1} c_2^{x_2}$. 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.

For $i = 1, 2$, we define the two functions $f_i(c_i, d_i) = \frac{c_i^{tx_i+y_i}}{d_i}$. Then $f_1(c_1, d_1) = f_2(c_2, d_2) = 1$ if and only if the ciphertext is consistent. Hence, the key K can be alternatively computed by first uniform $s_1, s_2 \in \mathbb{Z}_q$ and then

$$\begin{aligned} K &= c_1^{x_1} c_2^{x_2} \cdot (f_1(c_1, d_1))^{s_1} \cdot (f_2(c_2, d_2))^{s_2} \\ &= c_1^{x_1} c_2^{x_2} \cdot \left(\frac{c_1^{tx_1+y_1}}{d_1}\right)^{s_1} \cdot \left(\frac{c_2^{tx_2+y_2}}{d_2}\right)^{s_2} \\ &= \frac{c_1^{x_1+s_1(tx_1+y_1)} c_2^{x_2+s_2(tx_2+y_2)}}{d_1^{s_1} d_2^{s_2}} \end{aligned}$$

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 function f_1, f_2 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" $c_1^{x_1} \cdot c_2^{x_2}$.

5.2.4 Public Verification

In this section we show that in groups providing a DDH oracle DDHvf, consistency (or validity) of a given TBE ciphertext can be publicly verified. This is done 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 $c_i^{x_i t+y_i} = d_i$.

5.2.5 Security

Theorem 5.2 Under the gap decision linear assumption relative to generator $\mathcal{G}$, $\mathcal{LTBE}$ is selective-tag secure against chosen-ciphertext attacks.

Theorem 5.2 is proved in Appendix B.

The intuition of the proof is as follows. In the security reduction the DDH oracle provided by the Gap DLIN assumption is used to reject (as in the original scheme) every invalid ciphertext submitted by the adversary to the decryption oracle. Suppose we only want to show one-way security of $\mathcal{LTBE}$ i.e., the adversary's goal is to compute the challenge message (instead of deciding). The key idea of the reduction is based on an algebraic technique from [4]. An attacker $\mathcal{B}$ against the Gap DLIN problem can use the target tag t^* to setup the public-key for the adversary $\mathcal{A}$ attacking the security of $\mathcal{LTBE}$ in a way that (i) $\mathcal{B}$ (without knowing the secret key) can decrypt all ciphertexts with tag $t \neq t^*$; (ii) reconstructing the plaintext for a challenge ciphertext created with tag t^* can only be done by solving DLIN. If the adversary against $\mathcal{LTBE}$ is successfull so this adversary can be used to break Gap DLIN using the above simulation.

More details. Adversary $\mathcal{B}$ inputs a Gap DLIN instance $(g_1, g_2, z, g_1^{r_1^*}, g_2^{r_2^*}, w)$ and has to distinguish if $w = z^{r_1^*+r_2^*}$ or w=random. He picks a random values δ_1, δ_2 and defines the (correctly distributed) public key defined as $pk = (g_1, g_2, z, u_1 = z^{-t^*} g^{\delta_1}, u_2 = z^{-t^*} g^{\delta_2})$, where t^* is the target tag provided by $\mathcal{A}$.

Note that this way a consistent KEM ciphertext (c_1, c_2, d_1, d_2) for tag t properly created by the encryption algorithm has the form

$$c_i = g^{r_i}, \quad d_i = (z^t u_i)^r = (z^{r_i})^{t-t^*} c_i^{\delta_i} \quad (i = 1, 2)\ , \tag{1}$$

for some randomness $r_1, r_2 \in \mathbb{Z}_p$. Hence, in order to decrypt the challenge ciphertext $C^* = (c_1^*, c_2^*, d_1^*, d_2^*, \psi^*)$ for target tag t^* defined as

$$c_i^* = g_1^{r_i^*}, \quad d_i^* = (z^{r_i^*})^{t^*-t^*} c_i^{\delta_i} = c_i^{\delta_i} \quad (i = 1, 2)\ ,$$

(i.e., a ciphertext computed with unknown randomness r_1^*, r_2^* from the DLIN instance), adversary $\mathcal{A}$ has to compute the corresponding target key $K^* = z^{r_1^*+r_2^*}$ what is equivalent to breaking DLIN. On the other hand, for decrypting ciphertext $(c_1, c_2, d_1, d_2, \psi)$ for tag $t \neq t^*$, $\mathcal{B}$ first checks for consistency using the DDH oracle DDHvf provided by the Gap DLIN assumption. If the ciphertext is consistent the correct key $K = z^{r_1+r_2} = z^{r_1} \cdot z^{r_2}$ can be reconstructed by Equation (1) as $K = (d_1/c_1^{\delta_1})^{1/(t-t^*)} \cdot (d_2/c_2^{\delta_2})^{1/(t-t^*)}$ and hence the decryption query can be answered by computing $M = \psi \cdot K^{-1}$.

5.3 Efficiency

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. Note that the scheme does not make use of bilinear maps.

5.4 Based on the BDDH assumption

5.4.1 The BDDH assumption

The scheme will be parameterized by a *bilinear parameter generator*. This is a polynomial-time algorithm $\mathcal{G}_B$ that on input 1^k returns the description of a multiplicative cyclic group $\mathbb{G}$ of prime order p, where $2^k < p < 2^{k+1}$, the description of a multiplicative cyclic group $\mathbb{G}_T$ of the same order, a random element g generating $\mathbb{G}$, and a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. This bilinear pairing should be efficiently computable and satisfy the conditions below.

Bilinear: For all $g, h \in \mathbb{G}_1, x, y \in \mathbb{Z}$, $\hat{e}(g^x, h^y) = \hat{e}(g, h)^{xy}$

Non-degenerate: $\hat{e}(g, g) \neq 1_{\mathbb{G}_2}$

except the neutral element. Throughout the paper we use $\mathcal{BG} = (\mathbb{G}, \mathbb{G}_T, p, \hat{e}, g)$ (obtained by running $\mathcal{G}_B$) as shorthand for the description of bilinear groups.

Let $\mathcal{BG}$ be the description of bilinear groups and let $g \in \mathbb{G}_1$ be a random element from group $\mathbb{G}_1$. Consider the following problem formalized by Boneh and Franklin [7]: Given $(g, g^a, g^b, g^c, W) \in \mathbb{G}^4 \times \mathbb{G}_T$ as input, output yes if $W = \hat{e}(g, g)^{abc}$ and no otherwise. More formally we associate to an adversary $\mathcal{B}$ the following experiment:

Experiment $\mathbf{Exp}^{\text{bddh}}_{\mathcal{BG},\mathcal{B},(}(1^k)$
$a, b, c, w \xleftarrow{\$} \mathbb{Z}_p^*$
$\gamma \xleftarrow{\$} \{0, 1\}$; if $\gamma = 0$ then $w \leftarrow \hat{e}(g, g)^{abc}$ else $W \leftarrow \hat{e}(g, g)^w$
$\gamma' \xleftarrow{\$} \mathcal{B}(1^k, g, g^a, g^b, g^c, W)$
If $\gamma \neq \gamma'$ then return 0 else return 1

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

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

We say that the *Bilinear Decision Diffie-Hellman (BDDH) assumption relative to generator* $\mathcal{G}_B$ holds if $\mathbf{Adv}^{\text{bddh}}_{\mathcal{BG},\mathcal{B}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{B}$.

5.4.2 A TBE scheme based on BDDH

The following scheme is directly obtained by applying our generic transformation from IBE to TBE to an IBE scheme by Boneh and Boyen [4]. Let $\mathcal{BG}$ be a public description of a biliear group. We build a TBE scheme $\mathcal{BTBE} = (\mathsf{BTBE.kg}, \mathsf{BTBE.Enc}, \mathsf{BTBE.Dec})$ as follows.

$\mathsf{BTBE.kg}(1^k)$	$\mathsf{BTBE.Enc}(pk, t, M)$	$\mathsf{BTBE.Dec}(sk, t, C)$
$x_1, x_2, y \xleftarrow{\$} \mathbb{Z}_p^*$	$r \xleftarrow{\$} \mathbb{Z}_p^*$	Parse C as (c, d, ψ)
$u_1 \leftarrow g^{x_1}$; $u_2 \leftarrow g^{x_2}$; $v \leftarrow g^y$	$c \leftarrow g^r$; $d \leftarrow (u_1^t u_2)^r$	If $c^{x_1+tx_2} \neq d$ then reject
$z \leftarrow \hat{e}(g, v)$	$K \leftarrow z^r$; $\psi \leftarrow K \cdot M$	Else $K \leftarrow \hat{e}(c, v)$
$pk \leftarrow (u_1, u_2, z)$; $sk \leftarrow (x_1, x_2, v)$	$C \leftarrow (c, d, \psi)$	$M \leftarrow \psi \cdot K^{-1}$
Return (pk, sk)	Return $C \in \mathbb{G}^3$	Return M

The scheme's security is implied by Theorem 5.1 and the security results from [4].

Theorem 5.3 Under the bilinear decision Diffie-Hellman assumption relative to generator $\mathcal{G}$, $\mathcal{BTBE}$ is selective-tag secure against chosen-ciphertext attacks.

6 Direct Key Encpasulation

A *key encapsulation mechanism* [38] (KEM) $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Encaps}, \mathsf{KEM.Decaps})$ consits of three PTAs can be seen as a light PKE scheme. Instead of encrypting messages, the encapsulation algorithm $\mathsf{KEM.Encaps}$ 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 Appendix A.1.

Given a KEM and a DEM (aka symmetric encryption scheme), a hybrid public-key encryption scheme can be obtained by using the KEM to securely transport a random session key that is fed into the DEM to encrypt the plaintext message. It is well known that if both the KEM and the DEM are chosen-ciphertext secure, then the resulting hybrid encryption is also chosen-ciphertext secure [14, Sec. 7]. The security reduction is tight.

A DEM secure against chosen-ciphertext attacks can be built from relatively weak primitives, i.e. from any one-time symmetric encryption scheme by essentially adding a MAC. For concreteness we mention that a chosen-ciphertext secure PKE scheme can be built from our KEM construction with an additional overhead of a DEM which consists of a (one-time secure) symmetric encryption plus additional 128 bits for the MAC. Furthermore, Phan and Pointcheval [32] showed that *super pseudorandomn permutations* directly imply redundancy-free chosen-ciphertext secure DEMs that avoid the usual overhead due to the MAC. In practice, the modes of operation CMC [22] and EME [23] (provided that the underlying block-cipher is a strong pseudorandom permutation) can be used to encrypt large messages.

We note that for the natural task of securely generating a joint random session key, a KEM is sufficient and a fully-fledged public-key encryption scheme is not needed.

6.1 Based on the Linear Assumption

We build a KEM scheme as follows. Let $\mathsf{KEM.Kg}(1^k)$ be as in the TBE scheme of Section 5.2.2. The public key pk additionally contains a target collision resistant hash function $\mathrm{tcr} : \mathbb{G}\times\mathbb{G} \to \mathbb{Z}_q$ (i.e. given $t = \mathrm{tcr}(g_1, g_2)$ it should be hard to find $(h_1, h_2) \in \mathbb{G} \times \mathbb{G} \setminus \{(g_1, g_2)\}$ such that $\mathrm{tcr}(h_1, h_2) = t$; we refer to [13] for a formal definition).[5] The encapsulation/decapsulation algorithms are as follows:

$\mathsf{KEM.Encaps}(pk)$	$\mathsf{KEM.Decaps}(sk, c)$
$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$	Parse c as (c_1, c_2, d_1, d_2)
$c_1 \leftarrow g_1^{r_1}$; $c_2 \leftarrow g_2^{r_2}$	$t \leftarrow \mathrm{tcr}(c_1, c_2)$
$t \leftarrow \mathrm{tcr}(c_1, c_2)$	$s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p^*$
$d_1 \leftarrow z^{tr_1} u_1^{r_1}$; $d_2 \leftarrow z^{tr_2} u_2^{r_2}$	$K \leftarrow \frac{c_1^{x_1+s_1(tx_1+y_1)} \cdot c_2^{x_2+s_2(tx_2+y_2)}}{d_1^{s_1} \cdot d_2^{s_2}}$
$K \leftarrow z^{r_1+r_2}$	Return K
$c \leftarrow (c_1, c_2, d_1, d_2)$	
Return (c, K)	

Analogous to the TBE construction from Section 5.2 consistency of a ciphertext $c = (c_1, c_2, d_1, d_2)$ can be publicly verified by computing $t \leftarrow \mathrm{tcr}(c_1, c_2)$ and checking if $(g_i, z^t u_i, c_i, d_i)$ is a Diffie-

[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.

Hellman tuple for $i = 1, 2$.

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

The security reduction is tight and compared to the reduction from Theorem 5.2 there appears an additional additive factor taking into account a possible collision in the hash function tcr. The proof of Theorem 6.1 is similar to that of Theorem 5.2 and is given in Appendix C.

The way we use the target collision hash function is reminiscent to the Cramer-Shoup cryptosystem [13]. 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 $t^* = \text{tcr}(c_1^*, c_2^*)$. Consider a ciphertext (c_1, c_2, d_1, d_2) queried by adversary $\mathcal{A}$ during the CCA experiment and let $t = \text{tcr}(c_1, c_2)$. Similar to the proof of Theorem 5.2 we can setup the public-key in a way such that $\mathcal{B}$ is able to correctly 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.

6.2 Key Encapsulation based on the BDDH

We build a KEM scheme as follows. Let $\mathsf{KEM.Kg}(1^k)$ be as in the TBE scheme of Section 5.2.2. The encapsulation/decapsulation algorithms are defined as follows.

$\mathsf{KEM.Encaps}(pk)$	$\mathsf{KEM.Decaps}(sk, c)$
$r \xleftarrow{\$} \mathbb{Z}_p^*$	Parse c as $(c, d) \in \mathbb{G}^2$
$c \leftarrow g^r$; $t \leftarrow \text{tcr}(c)$	$t \leftarrow \text{tcr}(c)$
$d \leftarrow (u_1^t u_2)^r$	If $c^{xt+y} \neq d$ then $\perp$
$K \leftarrow z^r$	Elese return $K \leftarrow \hat{e}(c, v)$
$c \leftarrow (c, d)$	
Return (c, K)	

Theorem 6.2 Assume tcr is a target collision resistant hash function. Under the BDDH assumption relative to the generator $\mathcal{G}$ the above KEM is secure against chosen-ciphertext attacks.

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 [13] and the Kurosawa-Desmedt scheme [25] 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). Our BDDH-based KEM BMW from Section 6.2 equals the KEM by Boyen, Mei, and Waters [10]. 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.

Scheme	Origin	Assumption	Encryption	Decryption	Ciphertext	Public
			#pairings + #[multi,reg,fix]-exp		Overhead	Vfy?
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	—
DLIN-TBE (§5.2)	BK/TBE	DLIN	$0 + [2, 3, 0]$	$0 + [1, 0, 0]$	$4\|p\|$+com+mac	—
DLIN-KEM (§6.1)	KEM	DLIN	$0 + [2, 3, 0]$	$0 + [1, 0, 0]$	$4\|p\|$	yes
BDDH-TBE (§5.4)	BK/TBE	BDDH	$0 + [1, 2, 0]$	$1 + [0, 1, 0]$	$2\|p\|$+com+mac	—
BDDH-KEM (§6.2)	KEM	BDDH	$0 + [1, 2, 0]$	$1 + [0, 1, 0]$	$2\|p\|$	yes

Figure 2: Efficiency comparison for CCA-secure PKE schemes. Some figures are borrowed from [8, 6, 10]. 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 [36], multi-exponentiation ≈ 1.5, regular exponentiation = 1, fixed-base exponentiation $\ll 0.2$.

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.

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 5.2.1 already comes with a ciphertext overhead of $2|p|$.

In contrast to the comparison tables given in [6, 10] we do not consider the public-key of the recipient in the encryption algorithm as fixed. For that reason we do not count the exponentiations as (more efficient) fixed-base exponentiations. We think that this models more the typical goal of PKE schemes/KEMs since usually you send ciphertexts to many different recipients. If one really considers the receiver's public-key as fixed we can use an on-line/offline approach in all schemes as follows: the sender pre-computes and buffers some number of random (symmetric) keys. At the time the actual message is sent the sender simply looks up one of the buffered keys and encrypts the message with this key. In that case the online-part of the encryption algorithm is basically for free.

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 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 [25] 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 5.2 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] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 128–146, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 2.)

[2] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 1.)

[3] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 2, 3, 4, 10, 16.)

[4] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 13, 14.)

[5] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 3, 11.)

[6] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan 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. (Cited on page 9, 17.)

[7] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. (Cited on page 11, 14, 21.)

[8] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103, San Francisco, CA, USA, February 14–18, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 2, 8, 9, 17.)

[9] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *ACM CCS 04*, pages 168–177, Washington D.C., USA, October 25–29, 2004. ACM Press. (Cited on page 11.)

[10] Xavier Boyen, Qixiang Mei, and Brent 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. (Cited on page 4, 16, 17.)

[11] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *30th ACM STOC*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press. (Cited on page 1.)

[12] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 2, 5, 8, 9, 10, 21, 22.)

[13] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25, Santa Barbara, CA, USA, August 23–27, 1998. Springer-Verlag, Berlin, Germany. (Cited on page 2, 4, 15, 16.)

[14] Ronald Cramer and Victor 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. (Cited on page 2, 15.)

[15] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1978. (Cited on page 1.)

[16] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. (Cited on page 2.)

[17] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18, Santa Barbara, CA, USA, August 19–23, 1985. Springer-Verlag, Berlin, Germany. (Cited on page 1.)

[18] Edith Elkind and Amit 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/`. (Cited on page 2.)

[19] David Galindo and Ichiro Hasuo. Security notions for identity based encryption. Cryptology ePrint Archive, Report 2005/253, 2005. `http://eprint.iacr.org/`. (Cited on page 7.)

[20] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004. (Cited on page 9.)

[21] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984. (Cited on page 1.)

[22] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499, Santa Barbara, CA, USA, August 17–21, 2003. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[23] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 292–304, San Francisco, CA, USA, February 23–27, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[24] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. TCC 2006, 2006. (Cited on page 3.)

[25] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 2, 4, 16, 17.)

[26] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, october 1979. (Cited on page 9.)

[27] Philip D. MacKenzie, Michael K. Reiter, and Ke Yang. Alternatives to non-malleability: Definitions, constructions, and applications. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 171–190, Cambridge, MA, USA, February 19–21, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 2, 5, 6, 7, 8, 10.)

[28] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43, Seattle, Washington, USA, May 15–17, 1989. ACM Press. (Cited on page 23.)

[29] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. (Cited on page 1.)

[30] Lan Nguyen and Reihaneh Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 372–386, Jeju Island, Korea, December 5–9, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 11.)

[31] Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118, Cheju Island, South Korea, February 13–15, 2001. Springer-Verlag, Berlin, Germany. (Cited on page 3.)

[32] Duong Hieu Phan and David Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In Helena Handschuh and Anwar Hasan, editors, *SAC 2004*, volume 3357 of *LNCS*, Waterloo, Ontario, Canada, August 9–10, 2004. Springer-Verlag, Berlin, Germany. (Cited on page 15.)

[33] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444, Santa Barbara, CA, USA, August 11–15, 1991. Springer-Verlag, Berlin, Germany. (Cited on page 1, 5.)

[34] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978. (Cited on page 1.)

[35] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. (Cited on page 9, 23.)

[36] Michael Scott. Faster pairings using an elliptic curve with an efficient endomorphism. Cryptology ePrint Archive, Report 2005/252, 2005. `http://eprint.iacr.org/`. (Cited on page 17.)

[37] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1985. Springer-Verlag, Berlin, Germany. (Cited on page 21.)

[38] Victor Shoup. A proposal for an ISO standard for public key encryption (version 2.1). manuscript, 2001. Available on `http://shoup.net/papers/`. (Cited on page 3, 6, 7, 15, 21.)

[39] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany. (Cited on page 4.)

A Standard Definitions

A.1 Public Key Encapsulation Schemes

A *public-key encapsulation* (KEM) scheme [38] $\mathcal{KEM} = (\mathsf{KEM.Kg}, \mathsf{KEM.Encaps}, \mathsf{KEM.Decaps})$ with key-space $KeySp(k)$ consists of three PTAs. Via $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $(K, C) \xleftarrow{\$} \mathsf{KEM.Encaps}(1^k, pk)$ a key $K \in KeySp(k)$ together with a ciphertext C is created; via $K \leftarrow \mathsf{KEM.Decaps}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a key. For consistency, we require that for all $k \in \mathbb{N}$, and all $(K, C) \xleftarrow{\$} \mathsf{KEM.Encaps}(1^k, pk)$ we have $\Pr[\,\mathsf{KEM.Decaps}(C) = K\,] = 1$, where the probability is taken over the choice of $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(1^k)$, and the coins of all the algorithms in the expression above.

Definition A.1 Formally, we associate to an adversary $\mathcal{A}$ the following experiment:

$$
\begin{array}{l}
\mathbf{Exp}^{\textit{kem}\text{-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k) \\
\quad (pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(1^k) \\
\quad K_0^* \xleftarrow{\$} KeySp(k)\,;\ (K_1^*, C^*) \xleftarrow{\$} \mathsf{KEM.Encaps}(pk) \\
\quad b \xleftarrow{\$} \{0,1\} \\
\quad b \xleftarrow{\$} \mathcal{A}^{\textsc{Decaps}}(pk, K_b^*, C^*) \\
\quad \text{If } b \neq b \text{ then return 0 else return 1}
\end{array}
$$

where the oracle $\textsc{Decaps}(C)$ is returns $K \xleftarrow{\$} \mathsf{KEM.Decaps}(sk, C)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\textsc{Decaps}(\cdot)$ for the target ciphertext C^*. We define the advantage of $\mathcal{A}$ in the CCA experiment as

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

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

A.2 Identity Based Encryption

An *identity-based encryption* (IBE) scheme [37, 7, 12] $\mathcal{IBE} = (\mathsf{IBE.kg}, \mathsf{IBE.Keyder}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ consists of four PTAs. Via $(pk, sk) \xleftarrow{\$} \mathsf{IBE.kg}(1^k)$ the randomized key-generation algorithm produces master keys for security parameter $k \in \mathbb{N}$; via $sk[id] \xleftarrow{\$} \mathsf{IBE.Keyder}(sk, id)$

the master computes the secret key for identity id; via $C \xleftarrow{\$} \mathsf{IBE.Enc}(pk, id, M)$ a sender encrypts a message M to identity id to get a ciphertext; via $M \leftarrow \mathsf{IBE.Dec}(sk, C)$ 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 $k \in \mathbb{N}$, all identities id and messages $M \in MsgSp(k)$ we have $\Pr[\mathsf{IBE.Dec}(\mathsf{IBE.Keyder}(sk, id), \mathsf{IBE.Enc}(pk, id, M)) = M] = 1$, where the probability is taken over the choice of $(pk, sk) \xleftarrow{\$} \mathsf{IBE.kg}(1^k)$, and the coins of all the algorithms in the expression above.

Privacy. Following [12] we give the definition for selective-identity security of IBE schemes where an adversary has to commit to the target identity in advance. Let $\mathcal{IBE} = (\mathsf{IBE.kg}, \mathsf{IBE.Keyder}, \mathsf{IBE.Enc}, \mathsf{IBE}$ be an IBE scheme with associated message space $MsgSp$. To an adversary $\mathcal{A}$ and bit $b \in \{0, 1\}$ we associate the following experiment:

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\textit{ibe}\text{-sid-ind-cpa}}_{\mathcal{IBE},\mathcal{A}}(k) \\
\quad (id^*, St_0) \xleftarrow{\$} \mathcal{A}(1^k, \texttt{init}) \\
\quad (pk, sk) \xleftarrow{\$} \mathsf{IBE.kg}(1^k) \\
\quad (M_0^*, M_1^*, St) \xleftarrow{\$} \mathcal{A}^{\mathsf{IBE.Keyder}}(\texttt{find}, pk, St_0) \\
\quad b \xleftarrow{\$} \{0,1\} \,;\; C^* \xleftarrow{\$} \mathsf{IBE.Enc}(pk, id^*, M_b^*) \\
\quad b \xleftarrow{\$} \mathcal{A}^{\mathsf{IBE.Keyder}}(\texttt{guess}, C^*, St) \\
\quad \text{If } b \neq b \text{ then return 0 else return 1}
\end{array}
$$

where the oracle $\mathsf{IBE.Keyder}(id)$ is defined as

$$sk[id] \xleftarrow{\$} \mathsf{IBE.Keyder}(sk, id) \,;\;\; \text{Return } sk[id]$$

and $\mathcal{A}$ is not allowed to ask oracle $\mathsf{IBE.Keyder}(\cdot)$ for the target identity id^*. Both messages must be of the same size ($|M_0| = |M_1|$) and in the message space $MsgSp(k)$. We define the advantage of $\mathcal{A}$ in the corresponding experiment as

$$\mathbf{Adv}^{\textit{ibe}\text{-sid-ind-cpa}}_{\mathcal{IBE},\mathcal{A}}(k) \;=\; \left| \Pr\left[\mathbf{Exp}^{\textit{ibe}\text{-sid-ind-cpa}}_{\mathcal{IBE},\mathcal{A}}(k) = 1 \right] - \frac{1}{2} \right| .$$

IBE scheme $\mathcal{IBE}$ is said to be *selective-identity secure against chosen-plaintext attacks* if the advantage function $\mathbf{Adv}^{\textit{ibe}\text{-sid-ind-cpa}}_{\mathcal{IBE},\mathcal{A}}$ is a negligible function in k for all PTAs $\mathcal{A}$.

Similar to TBE we mention two known security variants of IBE schemes: (full) security against chosen-plaintext attacks (CPA-secure) and security against chosen-ciphertext attacks (CCA-secure). Let C^* be the target ciphertext and id^* be the target identity selected by the adversary $\mathcal{A}$ in the security experiment.

- To obtain the notion of *CPA security* for IBE schemes we modify the above security experiment in a way such that $\mathcal{A}$ does not have to commit to the target identity id^* in the beginning. Instead, $\mathcal{A}$ is allowed to choose id^* 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 *CCA-security*, we further modify the security experiment (CPA security) such that the we give the adversary furthermore access to an oracle answering all decryption queries suspect to $(id, C) \neq (id^*, C^*)$.

A.3 Target Collision Resistant Hash Functions

Let $(\mathrm{tcr}_s)_{s\in S}$ be a family of hash functions for security parameter k and with seed $s \in S = S(k)$. $\mathcal{F}$ is said to be *collision resistant* if, for a hash function $\mathrm{tcr} = \mathrm{tcr}_s$ (where the seed is chosen at random from S), it is infeasible for any polynomial-time adversary to find two distinct values $x \neq y$ such that $\mathrm{tcr}(x) = \mathrm{tcr}(y)$.

A weaker notion is that of *target collision resistant hash functions*. Here it should be infeasible for an polynomial-time adversary to find, given a randomly chosen element x and a randomly drawn hash function $\mathrm{tcr} = \mathrm{tcr}_s$, a distinct element $y \neq x$ such that $\mathrm{tcr}(x) = \mathrm{tcr}(y)$. (In collision resistant hash functions the value x may be chosen by the adversary.) Such hash functions are also called *universal one-way hash functions* [28] and can be built from arbitrary one-way functions [28, 35]. We define

$$\mathbf{Adv}^{\mathrm{cr}}_{\mathrm{tcr},\mathcal{A}}(k) = \Pr[\mathcal{A} \text{ finds a collision}].$$

Hash function family tcr is said to be a *target collision resistant* if the advantage function $\mathbf{Adv}^{\mathrm{cr}}_{\mathrm{tcr},\mathcal{A}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

B Proof of Theorem 5.2

Adversary $\mathcal{B}$ inputs an instance of the decisional linear problem, i.e. $\mathcal{B}$ inputs the values $(1^k, \mathbb{G}, g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w)$. $\mathcal{B}$'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 $\mathbf{Adv}^{\textit{tbe}\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)$. 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

$$\mathbf{Adv}^{\text{gap-dlin}}_{\mathcal{G},\mathcal{B}}(k) \geq \mathbf{Adv}^{\textit{tbe}\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)\ . \tag{2}$$

Now Eqn. (2) 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 $\mathcal{TBE} = (x_1, x_2, y_1, y_2)$ is known to $\mathcal{B}$.

Now consider an arbitrary ciphertext$C = (C_1, C_2, D_1, D_2)$ and let $t \in \mathbb{Z}_p$ be a tag. Recall that C 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

$$D_i = C_i^{tx_i+y_i} = C_i^{x_i t - t^* x_i + c_i} = (C_i^{x_i})^{t-t^*} \cdot C_i^{c_i}, \quad i \in \{1, 2\}\ . \tag{3}$$

By Equation (3), $D_i/C_i^{c_i} = (C_i^{x_i})^{t-t^*}$ and if $t \neq t^*$ then the session key $K = C_1^{x_1} \cdot C_2^{x_2}$ can alternatively be reconstructed as

$$K \leftarrow \left(\frac{D_1 \cdot D_2}{C_1^{c_1} \cdot C_2^{c_2}} \right)^{\frac{1}{t-t^*}} \quad . \tag{4}$$

Now adversary $\mathcal{B}$ runs $\mathcal{A}$ on input `find` and pk answering to its decryption queries as follows: Let $C = (C_1, C_2, D_1, D_2)$ be an arbitrary ciphertext submitted to the decryption oracle $\textsc{Dec}(C, t)$ for tag $t \neq t^*$. First $\mathcal{B}$ performs a public consistency check as explained in Section 5.2.4 using the Diffie-Hellman verification algorithm DDHvf. If C is not consistent then $\mathcal{B}$ returns a random message, as in the alternative (but equivalent) decryption algorithm (Section 5.2.3) of the original TBE scheme. Otherwise, if the ciphertext is consistent adversary $\mathcal{B}$ computes the session key by Equation (4) 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 $\textsc{Dec}(C, t)$ is identically distributed as the output of $\mathsf{TBE.Dec}(sk, C, t)$.

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 $C^* = (C_1^*, C_2^*, D_1^*, D_2^*, E^*)$ 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 (3), C^* 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 $\mathcal{TBE}$. 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^* 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 $|\Pr[b = b'] - 1/2| \geq \mathbf{Adv}^{tbe\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)$. On the other hand, when w is uniform in $\mathbb{G}$ then $\Pr[b = b'] = 1/2$. Therefore $\mathbf{Adv}^{\text{gap-dlin}}_{\mathcal{G},\mathcal{B}}(k) \geq \left| \left(\frac{1}{2} \pm \mathbf{Adv}^{tbe\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k) \right) - \frac{1}{2} \right| = \mathbf{Adv}^{tbe\text{-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(k)$. This proves Equation (2) and concludes the proof.

C Proof of Theorem 6.1

The proof is similar to that of Theorem 5.2. We point out the differences.

Adversary $\mathcal{B}$ inputs an instance of the decision linear problem, i.e. $\mathcal{B}$ inputs the values $(1^k, \mathbb{G}, g_1, g_2, z, g_1^{r_1}, g_2^{r_2}, w)$. Furthermore, $\mathcal{B}$ inputs the description of a target collision resistant hash function tcr. $\mathcal{B}$'s goal is to determine if $w = z^{r_1+r_2}$ or if w is a random group element or to find a collision in the hash function.

In the beginning of the simulation adversary $\mathcal{B}$ computes the target tag itself by setting $t^* = \mathcal{H}(g_1^{r_1}, g_2^{r_2})$. Depending on t^* the public key is computed as in the proof of Theorem 5.2. The challenge ciphertext c^* is computed as $c^* = (C_1^*, C_2^*, D_1^*, D_2^*) \leftarrow (g_1^{r_1}, g_2^{r_2}, (g_1^{r_1})^{c_1}, (g_2^{r_2})^{c_2})$, and the target key as $K^* = w$. With the same arguement as in the proof of Theorem 5.2 it can be shown that c is always consistent with target tag $t^* = \mathrm{tcr}(C_1^*, C_2^*)$ and when $w = z^{r_1+r_2}$, then $K^* = w$ is a correct key of ciphertext c^*.

Adversary $\mathcal{B}$ runs $\mathcal{A}(pk, K^*, c^*)$ answering to $\mathcal{A}$'s decryption queries as follows:
Let $c = (C_1, C_2, D_1, D_2)$ be a KEM ciphertext queried to the decapsulation oracle and let $t = \mathrm{tcr}(C_1, C_2)$ be its corresponding tag. Assume the ciphertext is consistent with tag t, otherwise $\mathcal{B}$ returns a random key K. Now adversary $\mathcal{B}$ has to distinguish three cases:

Case 1: $t \neq t^*$**.** Computation of the key K can be simulated as in the proof of Theorem 5.2.

Case 2: $t = t^*$ **and** $(C_1, C_2) \neq (C_1^*, C_2*)$**.** In this case $\mathcal{B}$ has found a collision in the hash function tcr, i.e. we have $\mathrm{tcr}(C_1, C_2) = \mathrm{tcr}(C_1^*, C_2^*)$ for distinct inputs to the hash function.[6]

Case 3: $t = t^*$ **and** $(C_1, C_2) = (C_1^*, C_2^*)$**.** Then ciphertext is consistent with tag t^* if and only if $D_1 = D_1^*$ and $D_2 = D_2^*$ (this is since for fixed C_1, C_2 and t and assuming the ciphertext is consistent with t, the values D_1, D_2 are uniquely defined). Then we have $c = c^*$ and $\mathcal{B}$ returns $\perp$ since in this case $\mathcal{A}$ queried for the target ciphertext c^*.

Note that this perfectly simulates the view of adversary $\mathcal{A}$ as in the real experiment.

The probability that $\mathcal{B}$ finds a collision in the hash function tcr is $\mathbf{Adv}^{\mathrm{cr}}_{\mathrm{tcr},\mathcal{H}}(k)$. Assume there were no hash collision found by $\mathcal{B}$. When the value w input by $\mathcal{H}$ 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 $|\Pr[b = b'] - 1/2| > \mathbf{Adv}^{\mathit{kem}\text{-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k)$. On the other hand, when w is uniform in $\mathbb{G}$ then $\Pr[b = b'] = 1/2$. From the above we summarize that the success probability of $\mathcal{B}$ breaking the decisional linear assumption is bounded as as follows:

$$\mathbf{Adv}^{\text{gap-dlin}}_{\mathcal{G},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathit{kem}\text{-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k) - \mathbf{Adv}^{\mathrm{cr}}_{\mathrm{tcr},\mathcal{H}}(k)\ , \tag{5}$$

where $\mathbf{Adv}^{\mathrm{cr}}_{\mathrm{tcr},\mathcal{H}}(k)$ is the security of the target collision resistant hash function.

[6] Note that since $(C_1^*, C_2^*) = (g_1^{r_1}, g_2^{r_2})$ were chosen at random from the outside of the experiment this contradicts the *target* collision resistance of tcr.