Chosen-Ciphertext Security from Identity-Based Encryption

Ran Canetti[*] Shai Halevi[*] Jonathan Katz[†]

September 2, 2003

Abstract

We show how to construct a CCA-secure public-key encryption scheme from any CPA-secure identity-based encryption (IBE) scheme. Our conversion from IBE to a CCA-secure scheme is simple, efficient, and provably secure in the standard model (i.e., security of the resulting scheme does not rely on the random oracle model). In addition, the resulting scheme achieves CCA security even if the underlying IBE scheme satisfies only a "weak" notion of security which is known to be achievable in the standard model based on the bilinear Diffie-Hellman assumption. Thus, our results yield a new construction of CCA-secure public-key encryption in the standard model. Interestingly, the resulting scheme avoids any non-interactive proofs of "well-formedness" which were shown to underlie all previously-known constructions.

We also extend our technique to obtain a simple and reasonably efficient method for securing any BTE scheme against adaptive chosen-ciphertext attacks. This, in turn, yields more efficient constructions of CCA-secure (hierarchical) identity-based and forward-secure encryption schemes in the standard model.

Our results — building on previous black-box separations — rule out black-box constructions of IBE from CPA-secure public-key encryption.

Keywords: Chosen-ciphertext security, Forward-secure encryption, Identity-based encryption, Public-key encryption.

[*] IBM T.J. Watson Research Center, NY, USA. {canetti,shaih}@watson.ibm.com.

[†] Department of Computer Science, University of Maryland (College Park). jkatz@cs.umd.edu.

1 Introduction

Security against adaptive chosen-ciphertext attacks (i.e., "CCA security") [19, 9, 1] is a strong and very useful notion of security for public-key encryption schemes. This notion is known to suffice for many applications of encryption in the presence of *active* attackers, including secure communication, auctions, voting schemes, and many others. Indeed, CCA security is commonly accepted as the security notion of choice for encryption schemes that are to be "plugged in" to a protocol running in an arbitrary setting; see, e.g., [22, 23].

However, there are only a handful of known public-key encryption schemes that can be proven to be CCA-secure in the standard model (i.e., without the use of heuristics such as random oracles). In fact, only two main techniques have been proposed for constructing such cryptosystems. The first follows the paradigm of Naor and Yung [18], as extended by Dolev, Dwork, and Naor [9] and later simplified by Sahai [20] and Lindell [17]. This technique uses as building blocks any CPA-secure public-key encryption scheme (i.e., any scheme that is secure against chosen-plaintext attacks [14]) as well as any non-interactive zero-knowledge (NIZK) proof system [3, 11] (which can be constructed using any family of trapdoor permutations). The resulting encryption scheme, however, is highly inefficient precisely because it employs an NIZK proof, which in turn uses a generic Karp reduction from an instance of the encryption scheme to an instance of some NP-complete problem; note further that there are no known efficient NIZK proof systems even under specific assumptions and for particular cryptosystems of interest. Thus, this general technique for constructing CCA-secure cryptosystems should be viewed more as a "proof of feasibility" than as a practical construction.

The second technique is due to Cramer and Shoup [7, 8], and is based on algebraic constructs with some specific homomorphic properties (namely, those which admit "smooth hash proof systems"; see [8]). Algebraic constructs of the appropriate type are known to exist based on some specific assumptions: the hardness of the decisional Diffie-Hellman problem [7] or the hardness of deciding quadratic residuosity or N^{th} residuosity in certain groups [8]. This technique leads to a number of instantiations which are efficient enough to be used in practice.

Interestingly, as observed by Elkind and Sahai [10], both of these techniques for constructing CCA-secure encryption schemes can be viewed as special cases of a *single* paradigm. In this more general paradigm (informally), one starts with a CPA-secure cryptosystem in which certain "ill-formed" ciphertexts are indistinguishable from "well-formed" ones. A CCA-secure cryptosystem is then obtained by having the sender include a "proof of well-formedness" for the transmitted ciphertext. Both NIZK proofs and smooth hash proof systems were shown to meet the requirements for these proofs of well-formedness.

1.1 Our contributions

We present here a new construction of CCA-secure public-key encryption schemes based on any identity-based encryption (IBE) scheme satisfying a relatively "weak" notion of security. An IBE scheme meeting this notion of security in the standard model was recently proposed (see below); thus, our technique yields a new construction of CCA-secure encryption in the standard model. The resulting construction is very simple and reasonably efficient; in particular, it avoids “proofs of well-formedness” of any sort, hence it seems not to follow the methodology of Elkind and Sahai.

Before sketching our construction, we first recall the notion of IBE. The concept of identity-based encryption was introduced by Shamir [21], and provably-secure IBE schemes (in the random oracle model) were recently demonstrated by Boneh and Franklin [4] and Cocks [6]. An IBE scheme is a public-key encryption scheme in which, informally, any string (i.e., identity) can serve as a public key. In more detail, a trusted private-key generator (PKG) initializes the system by running a key-generation algorithm to generate “master” public and secret keys. The public key is published, while the PKG stores the secret key. Given any string $id \in \{0,1\}^*$ (which can be viewed as a receiver's identity), the PKG can derive a “personal secret key” SK_{id}. Any sender can encrypt a message for this receiver using only the master public key and the string id. The resulting ciphertext can be decrypted using the derived secret key SK_{id}, but the message remains hidden from an adversary who does not know SK_{id} even if that adversary is given $SK_{id'}$ for various identities $id' \neq id$.

In the definition of security for IBE given by Boneh and Franklin [4], the adversary is allowed to choose the “target identity” (id in the above discussion) in an adaptive manner, possibly based on the master public key and any keys $SK_{id'}$ the adversary has obtained thus far. Boneh and Franklin construct a scheme meeting this definition of security based on the bilinear Diffie-Hellman (BDH) assumption in the random oracle model. A weaker notion of security for IBE, introduced by Canetti, Halevi and Katz [5], requires the adversary to specify the target identity *before* the public-key is published; we will refer to this notion of security as “weak” IBE. It has been shown [5] (building on earlier work of Gentry and Silverberg [12]) that a weak IBE scheme can be constructed based on the BDH assumption *in the standard model*.

Our construction of CCA-secure encryption requires only an IBE scheme satisfying this weaker notion of security. The conversion of any such IBE scheme to a CCA-secure (standard) public-key encryption scheme proceeds as follows: The public key of the new scheme is simply the master public key of the IBE scheme, and the secret key is the corresponding master secret key. To encrypt a message, the sender first generates a key-pair (vk, sk) for some one-time signature scheme, and then encrypts the message with respect to the “identity” vk. The resulting ciphertext c is then signed using sk to obtain a signature σ. The final ciphertext consists of the verification key vk, the IBE ciphertext c, and the signature σ. To decrypt the ciphertext $\langle vk, c, \sigma \rangle$, the receiver first verifies the signature on c with respect to vk (and outputs $\perp$ if the signature is not correct). The receiver then derives the secret key SK_{vk} corresponding to the “identity” vk, and uses SK_{vk} to decrypt the ciphertext c as per the underlying IBE scheme.

Security of the above scheme against adaptive chosen-ciphertext attacks can be informally understood as follows. Say a ciphertext $\langle vk, c, \sigma \rangle$ is *valid* if σ is a valid signature on c with respect to vk. Now consider a “challenge ciphertext” $C = \langle vk, c, \sigma \rangle$ given to the adversary. Any valid ciphertext $C' = \langle vk', c', \sigma' \rangle$ submitted by the adversary to a decryption oracle (implying $C' \neq C$), must have $vk' \neq vk$ by the security of the one-time signature scheme. But then the crux of the security proof reduces to showing that (weak) security of the IBE scheme implies that decrypting C' does not give the adversary any further advantage in decrypting the challenge ciphertext. Intuitively, this is because the adversary would be unable to decrypt the underlying ciphertext c *even if it had the secret key* $SK_{vk'}$

corresponding to vk' (since $vk' \neq vk$, and c was formed using an IBE scheme).

Further extensions and applications. Canetti, Halevi, and Katz [5] also introduced the notion of binary tree encryption (BTE), and showed how to construct a secure BTE scheme in the standard model. They also show how to construct both hierarchical IBE (HIBE) schemes [15, 12] and forward-secure encryption (FSE) schemes starting from any BTE scheme, again in the standard model. To handle security against chosen-ciphertext attacks for each of these constructions, they suggest using the general technique of Naor and Yung [18] as adapted by Sahai and Lindell [20, 17]. This involves the use of NIZK proofs, as noted above, which makes the resulting CCA-secure schemes highly inefficient.

Here, we extend our technique to obtain a simple conversion from any semantically-secure BTE scheme to a CCA-secure BTE scheme. The resulting BTE scheme is considerably more efficient than a BTE scheme derived using the previously-suggested approach (based on NIZK); furthermore, the efficiency gain carries over immediately to yield improved constructions of CCA-secure HIBE and FSE schemes as well.

A "black-box separation" between CPA-secure encryption and weak IBE. Our construction of a CCA-secure encryption scheme from any IBE scheme is *black box* in the sense that it only uses the underlying IBE scheme by invoking its prescribed interface (and not, for example, by using the circuit which implements the IBE scheme). Recently, Gertner, Malkin, and Myers [13] have shown (in particular) that there do not exist black-box constructions of CCA-secure public-key encryption schemes from CPA-secure ones. Combined with their result, the results in the current work give a black-box separation between CPA-secure encryption and weak IBE; in other words, there are no black-box constructions of the latter from the former.

Although a result of this sort should not be viewed as a strict impossibility result (after all, the known constructions of CCA-secure encryption schemes based on trapdoor permutations [9, 20] rely on NIZK and are therefore *non*-black box), it does rule out certain techniques for constructing IBE schemes based on general assumptions.

2 Definitions

We review the standard definitions of public-key encryption schemes and their security against adaptive chosen-ciphertext attacks [1]. This is followed by a definition of identity-based encryption (IBE) schemes [4] and binary tree encryption (BTE) schemes [5] and their security against chosen-plaintext attacks (following [4, 5]).

2.1 Public-Key Encryption

Definition 1 A public-key encryption scheme PKE is a triple of PPT algorithms $(\mathsf{Gen}, \mathcal{E}, \mathcal{D})$:

- The randomized *key generation algorithm* Gen takes as input a security parameter 1^k and outputs a public key PK and a secret key SK. We write $(PK, SK) \leftarrow \mathsf{Gen}(1^k)$.

- The randomized *encryption algorithm* $\mathcal{E}$ takes as input a public key PK and a message $m \in \{0,1\}^*$, and outputs a ciphertext C. We write $C \leftarrow \mathcal{E}_{PK}(m)$.

- The *decryption algorithm* $\mathcal{D}$ takes as input a ciphertext C and a secret key SK. It returns a message $m \in \{0,1\}^*$ or the distinguished symbol $\perp$. We write $m \leftarrow \mathcal{D}_{SK}(C)$.

We require that for all (PK, SK) output by Gen, all $m \in \{0,1\}^*$, and all C output by $\mathcal{E}_{PK}(m)$ we have $\mathcal{D}_{SK}(C) = m$. ■

We recall the standard definition of security for public-key encryption schemes against adaptive chosen-ciphertext attacks.

Definition 2 A PKE scheme is secure against adaptive chosen-ciphertext attacks (i.e., "CCA-secure") if the advantage of any PPT adversary A in the following game is negligible in the security parameter k:

1. $\mathsf{Gen}(1^k)$ outputs (PK, SK). Adversary A is given 1^k and PK.
2. The adversary may make polynomially-many queries to a decryption oracle $\mathcal{D}_{SK}(\cdot)$.
3. At some point, A outputs two messages m_0, m_1 with $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $C^* \leftarrow \mathcal{E}_{PK}(m_b)$.
4. A may continue to query its decryption oracle $\mathcal{D}_{SK}(\cdot)$ except that it may not request the decryption of C^*.
5. Finally, A outputs a guess b'.

We say A *succeeds* if $b' = b$, and denote the probability of this event by $\Pr_{A,\mathsf{PKE}}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $|\Pr_{A,\mathsf{PKE}}[\mathsf{Succ}] - 1/2|$. ■

2.2 Identity-Based Encryption

In an IBE scheme, an arbitrary identity (i.e., bit string) can serve as a public key once some master parameters have been established by a (trusted) private key generator (PKG). We review the definition of Boneh and Franklin [4].

Definition 3 An identity-based encryption scheme IBE is a 4-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Der}, \mathcal{E}, \mathcal{D})$ such that:

- The randomized *setup algorithm* Setup takes as input a security parameter 1^k and a value ℓ for the identity length. It outputs some system-wide parameters PK along with a master secret key msk. (We assume that k and ℓ are implicit in PK.)

- The (possibly randomized) *key derivation algorithm* Der takes as input the master key msk and an identity $ID \in \{0,1\}^\ell$. It returns the corresponding decryption key SK_{ID}. We write $SK_{ID} \leftarrow \mathsf{Der}_{\mathsf{msk}}(ID)$.

- The randomized *encryption algorithm* $\mathcal{E}$ takes as input the system-wide public key PK, an identity $ID \in \{0,1\}^\ell$, and a message $m \in \{0,1\}^*$, and outputs a ciphertext C. We write $C \leftarrow \mathcal{E}_{PK}(ID, m)$.

- The *decryption algorithm* $\mathcal{D}$ takes as input an identity ID, its associated decryption key SK_{ID}, and a ciphertext C. It outputs a message $m \in \{0,1\}^*$ or the distinguished symbol $\perp$. We write $m \leftarrow \mathcal{D}_{SK_{ID}}(ID, C)$.

We require that for all (PK, msk) output by Setup, all $ID \in \{0,1\}^\ell$, all SK_{ID} output by $\mathsf{Der}_{\mathsf{msk}}(ID)$, all $m \in \{0,1\}^*$, and all C output by $\mathcal{E}_{PK}(ID, m)$ we have $\mathcal{D}_{SK_{ID}}(ID, C) = m$. ■

We now give a definition of security for IBE. As mentioned earlier, this definition is weaker than that given by Boneh and Franklin and conforms to the "selective-node" attack considered by Canetti, et al. [5]. Under this definition, the identity for which the challenge ciphertext is encrypted is selected by the adversary *in advance* (i.e., "non-adaptively") before the public key is generated. Since an IBE scheme satisfying this definition suffices for our purposes, this only makes our results stronger. Furthermore, a scheme satisfying this definition of security in the standard model is known [5]. (For the case of the original definition of Boneh and Franklin, only constructions in the random oracle model are known.)

Definition 4 An IBE scheme is secure against selective-identity, chosen-plaintext attacks if for all polynomially-bounded functions $\ell(\cdot)$, the advantage of any PPT adversary A in the following game is negligible in the security parameter k:

1. $A(1^k, \ell(k))$ outputs a target identity $ID^* \in \{0,1\}^{\ell(k)}$.
2. $\mathsf{Setup}(1^k, \ell(k))$ outputs (PK, msk). The adversary is given PK.
3. The adversary A may make polynomially-many queries to an oracle $\mathsf{Der}_{\mathsf{msk}}(\cdot)$, except that it may not request the secret key corresponding to the target identity ID^*.
4. At some point, A outputs two messages m_0, m_1 with $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $C^* \leftarrow \mathcal{E}_{PK}(ID^*, m_b)$.
5. A may continue to query its oracle $\mathsf{Der}_{\mathsf{msk}}(\cdot)$, but still may not request the secret key corresponding to the identity ID^*.
6. Finally, A outputs a guess b'.

We say A *succeeds* if $b' = b$, and denote the probability of this event by $\Pr_{A,\mathsf{IBE}}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $|\Pr_{A,\mathsf{IBE}}[\mathsf{Succ}] - 1/2|$. ■

One may extend the above definition to consider security against selective-identity, *(adaptive) chosen-ciphertext attacks.* In this case, the above definition is extended largely as one might expect: in addition to the game as outlined above, the adversary now additionally has access to an oracle $\widehat{\mathcal{D}}(\cdot)$ such that $\widehat{\mathcal{D}}(C)$ returns $\mathcal{D}_{SK_{ID^*}}(C)$, where SK_{ID^*} is the secret key associated with the target identity ID^* (computed using $\mathsf{Der}_{\mathsf{msk}}(ID^*)$).[1] As usual, the adversary has access to this oracle throughout the entire game, but cannot submit the challenge ciphertext C^* to $\widehat{\mathcal{D}}$.

2.3 Binary Tree Encryption

Binary tree encryption (BTE) was introduced by Canetti, Halevi, and Katz [5], and may be viewed as a relaxed variant of hierarchical identity-based encryption (HIBE) [15, 12] in the

[1] Note that decryption queries for identities $ID' \neq ID^*$ are superfluous, as A may make the corresponding Der query itself and thereby obtain $SK_{ID'}$.

following sense: in a BTE scheme, each node has two children labeled "0" and "1", while in a HIBE scheme, each node has arbitrarily-many children labeled with arbitrary strings. Although BTE is therefore a weaker primitive, it is known [5] that a BTE scheme supporting a binary tree of depth polynomial in the security parameter may be used to construct a full-fledged HIBE scheme (and thus, in particular, an ID-based encryption scheme).

Definition 5 A binary tree encryption scheme BTE is a 4-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Der}, \mathcal{E}, \mathcal{D})$ such that:

- The randomized *setup algorithm* Setup takes as input a security parameter 1^k and a value ℓ representing the maximum tree depth. It outputs some system-wide parameters PK along with a master (root) secret key SK_ε. (We assume that k and ℓ are implicit in PK and all secret keys.)
- The (possibly randomized) *key derivation algorithm* Der takes as input the name of a node $w \in \{0,1\}^{<\ell}$ and its associated secret key SK_w. It returns secret keys SK_{w0}, SK_{w1} for the two children of w.
- The randomized *encryption algorithm* $\mathcal{E}$ takes as input PK, the name of a node $w \in \{0,1\}^{\leq \ell}$, and a message m, and returns a ciphertext C. We write $C \leftarrow \mathcal{E}_{PK}(w, m)$.
- The *decryption algorithm* $\mathcal{D}$ takes as input the name of a node $w \in \{0,1\}^{\leq \ell}$, its associated secret key SK_w, and a ciphertext C. It returns a message m or the distinguished symbol $\perp$. We write $m \leftarrow \mathcal{D}_{SK_w}(w, C)$.

We require that for all (PK, SK_ε) output by Setup, any $w \in \{0,1\}^{\leq \ell}$ and any correctly-generated secret key SK_w for this node, any message m, and all C output by $\mathcal{E}_{PK}(w, m)$ we have $\mathcal{D}_{SK_w}(w, C) = m$. ■

The following definition of security for BTE, due to [5], is weaker than the corresponding notion of security for HIBE given by Gentry and Silverberg [12]. As in the definition of security for ID-based encryption given previously, the following definition refers to a "non-adaptive" selection of the node for which the challenge ciphertext is encrypted. Again, however, this definition suffices for our application; furthermore, a construction meeting this definition of security in the standard model is known [5] (in contrast, a construction meeting the stronger security definition of [12] is known only in the random oracle model and only for trees of constant depth).

Definition 6 A BTE scheme is secure against selective-node, chosen-plaintext attacks if for all polynomially-bounded functions $\ell(\cdot)$, the advantage of any PPT adversary A in the following game is negligible in the security parameter k:

1. $A(1^k, \ell(k))$ outputs a node label $w^* \in \{0,1\}^{\leq \ell(k)}$.
2. $\mathsf{Setup}(1^k, \ell(k))$ outputs (PK, SK_ε). In addition, algorithm $\mathsf{Der}(\cdots)$ is used to generate the secret keys of all the nodes on the path from the root to w^*, and also the secret keys for the two children of w^* (if $|w^*| < \ell$). The adversary is given PK and the secret keys $\{SK_w\}$ for all nodes w of the following form:

– $w = w'\overline{b}$, where $w'b$ is a prefix of w^* and $b \in \{0,1\}$ (i.e., w is a sibling of some node in P);

– $w = w^*0$ or $w = w^*1$ (i.e., w is a child of w^*; this is only when $|w^*| < \ell$).

Note that this allows the adversary to compute $SK_{w'}$ for any node $w' \in \{0,1\}^{\leq \ell(k)}$ that is *not* a prefix of w^*.

3. At some point, A outputs two messages m_0, m_1 with $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $C^* \leftarrow \mathcal{E}_{PK}(w^*, m_b)$.

4. Finally, A outputs a guess b'.

We say that A *succeeds* if $b' = b$, and denote the probability of this event by $\Pr_{A,\mathsf{BTE}}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $|\Pr_{A,\mathsf{BTE}}[\mathsf{Succ}] - 1/2|$. ■

A BTE scheme meeting the above definition of security will be termed "secure in the sense of SN-CPA". We may also define the stronger notion of security against selective-node, *adaptive chosen-ciphertext attacks*. (We refer to scheme meeting this definition of security as "secure in the sense of SN-CCA".) Such a definition can be found in [5], and we describe it informally here: the above game is modified so that the adversary additionally has access to an oracle $\widehat{\mathcal{D}}$ such that $\widehat{\mathcal{D}}(w, C)$ first computes the secret key SK_w for node w (using SK_ε and repeated calls to to Der); the oracle then outputs $m \leftarrow \mathcal{D}_{SK_w}(w, C)$. The adversary has access to this oracle throughout the entire game, but may not query $\widehat{\mathcal{D}}(w^*, C^*)$ after receiving the challenge ciphertext C^* (we stress that the adversary is allowed to query $\widehat{\mathcal{D}}(w, C^*)$ for $w \neq w^*$, as well as $\widehat{\mathcal{D}}(w^*, C)$ for $C \neq C^*$).

3 Chosen-Ciphertext Security from ID-Based Encryption

Given an ID-based encryption scheme $\Pi' = (\mathsf{Setup}, \mathsf{Der}, \mathcal{E}', \mathcal{D}')$ secure against selective-identity chosen-plaintext attacks, we construct a (standard) public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathcal{E}, \mathcal{D})$ secure against chosen-ciphertext attacks. In the construction, we use a one-time signature scheme $\mathsf{Sig} = (\mathcal{G}, \mathsf{Sign}, \mathsf{Vrfy})$, in which the verification key output by $\mathcal{G}(1^k)$ has length $\ell_s(k)$. We need this scheme to be secure in the sense of *strong* unforgeability (i.e., an adversary is unable to forge even a new signature on the previously-signed message). We note that such a scheme may be based on any one-way function [16] so, in particular, such a scheme exists given the existence of Π'. The construction of Π proceeds as follows:

- $\mathsf{Gen}(1^k)$ runs $\mathsf{Setup}(1^k, \ell_s(k))$ to obtain (PK, msk). The public key is PK and the secret key is msk.

- To encrypt message m using public key PK, the sender first runs $\mathcal{G}(1^k)$ to obtain verification key vk and signing key sk (with $|vk| = \ell_s(k)$). The sender then computes $C \leftarrow \mathcal{E}'_{PK}(vk, m)$ (i.e., the sender encrypts m with respect to "identity" vk) and $\sigma \leftarrow \mathsf{Sign}_{sk}(C)$. The final ciphertext is $\langle vk, C, \sigma \rangle$.

- To decrypt ciphertext $\langle vk, C, \sigma \rangle$ using secret key msk, the receiver first checks whether $\mathsf{Vrfy}_{vk}(C, \sigma) \overset{?}{=} 1$. If not, the receiver simply outputs $\perp$. Otherwise, the receiver computes $SK_{vk} \leftarrow \mathsf{Der}_{\mathsf{msk}}(vk)$ and outputs $m \leftarrow \mathcal{D}'_{SK_{vk}}(ID, C)$.

We first give some intuition as to why Π is secure against chosen-ciphertext attacks. Let $\langle vk^*, C^*, \sigma^* \rangle$ be the challenge ciphertext (cf. Definition 2). It should be 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 Π' which is CPA-secure, vk^* is independent of the message, and σ^* is merely the result of applying the signing algorithm to C^*.

We claim that decryption oracle queries cannot further help the adversary is guessing the value of b. On one hand, if the adversary submits ciphertext $\langle vk', C', \sigma' \rangle$ different from the challenge ciphertext but with $vk' = vk^*$ then the decryption oracle will reply with $\perp$ since the adversary is unable to forge new, valid signatures with respect to vk. On the other hand, if $vk' \neq vk^*$ then (informally) the decryption query will not help the adversary since the eventual decryption using $\mathcal{D}'$ (in the underlying scheme Π') will be done with respect to a different "identity" vk'. Below, we formally prove that this cannot help an adversary.

Theorem 1 *If* Π' *is an* IBE *scheme which is secure against selective-identity, chosen-plaintext attacks and* Sig *is a strongly unforgeable one-time signature scheme, then* Π *is a* PKE *scheme which is secure against adaptive chosen-ciphertext attack.*

Proof Given any PPT adversary $\mathcal{A}$ attacking Π in an adaptive chosen-ciphertext attack, we construct a PPT adversary $\mathcal{A}'$ attacking Π' in a selective-identity, chosen-plaintext attack. Relating the success probabilities of these adversaries gives the desired result.

Before specifying $\mathcal{A}'$, we first define event Forge and bound the probability of its occurrence. Let $\langle vk^*, C^*, \sigma^* \rangle$ be the challenge ciphertext received by $\mathcal{A}$, and let Forge denote the event that $\mathcal{A}$ submits to its decryption oracle a ciphertext $\langle vk^*, C, \sigma \rangle$ with $(C, \sigma) \neq (C^*, \sigma^*)$ but for which $\mathsf{Vrfy}_{vk^*}(C, \sigma) = 1$. (We include in this event the case when $\mathcal{A}$ submits such a query to its decryption oracle *before* receiving the challenge ciphertext; in this case, we do not require $(C, \sigma) \neq (C^*, \sigma^*)$.) It is easy to see that we can use $\mathcal{A}$ to break the underlying one-time signature scheme Sig with probability exactly $\Pr_{\mathcal{A}}[\mathsf{Forge}]$; since Sig is a strongly unforgeable one-time signature scheme, it must be the case that $\Pr_{\mathcal{A}}[\mathsf{Forge}]$ is negligible (in the security parameter k).

We now define adversary $\mathcal{A}'$ as follows:

1. $\mathcal{A}'(1^k, \ell_s(k))$ runs $\mathcal{G}(1^k)$ to generate (vk^*, sk^*). It then outputs the "target identity" $ID^* = vk^*$.

2. $\mathsf{Setup}(1^k, \ell_s(k))$ outputs (PK, msk) and $\mathcal{A}'$ is given PK. Adversary $\mathcal{A}'$, in turn, runs $\mathcal{A}$ on input 1^k and PK.

3. When $\mathcal{A}$ makes decryption oracle query $\mathcal{D}(\langle vk, C, \sigma \rangle)$, adversary $\mathcal{A}'$ proceeds as follows:

   (a) If $\mathsf{Vrfy}_{vk}(C, \sigma) \neq 1$, then $\mathcal{A}'$ simply returns $\perp$.

   (b) If $\mathsf{Vrfy}_{vk}(C, \sigma) = 1$ and $vk = vk^*$ (i.e., event Forge occurs), then $\mathcal{A}'$ halts and outputs a random bit.

   (c) If $\mathsf{Vrfy}_{vk}(C, \sigma) = 1$ and $vk \neq vk^*$, then $\mathcal{A}'$ makes the oracle query $\mathsf{Der}_{\mathsf{msk}}(vk)$ to obtain SK_{vk}. It then computes $m \leftarrow \mathcal{D}'_{SK_{vk}}(vk, C)$ and returns m.

4. At some point, $\mathcal{A}$ outputs two equal-length messages m_0, m_1. These same messages are output by $\mathcal{A}'$. In return, $\mathcal{A}'$ is given a challenge ciphertext C^*; adversary $\mathcal{A}'$ then computes $\sigma^* \leftarrow \mathsf{Sign}_{vk^*}(C^*)$ and returns $\langle vk^*, C^*, \sigma^* \rangle$ to $\mathcal{A}$.

5. $\mathcal{A}$ may continue to make decryption oracle queries, and these are answered as before. (Recall, we assume that $\mathcal{A}$ does not query the decryption oracle on the challenge ciphertext itself.)

6. Finally, $\mathcal{A}$ outputs a guess b'; this same guess is output by $\mathcal{A}'$.

Note that $\mathcal{A}'$ represents a legal adversarial strategy for attacking Π' in a selective-identity, chosen-plaintext attack; in particular, $\mathcal{A}'$ never requests the secret key corresponding to "target identity" vk^*. Furthermore, $\mathcal{A}'$ provides a perfect simulation for $\mathcal{A}$ (and thus $\mathcal{A}'$ succeeds whenever $\mathcal{A}$ succeeds) unless event Forge occurs. We therefore have:

$$\Pr_{\mathcal{A}',\Pi'}[\mathsf{Succ}] \geq \Pr_{\mathcal{A},\Pi}[\mathsf{Succ}] - \tfrac{1}{2} \cdot \Pr_{\mathcal{A}}[\mathsf{Forge}].$$

Since $\Pr_{\mathcal{A}',\Pi'}[\mathsf{Succ}]$ is negligibly close to $1/2$ (because Π' is assumed to be secure in against selective-identity, chosen-plaintext attacks), it must be the case that $\Pr_{\mathcal{A},\Pi}[\mathsf{Succ}]$ is negligibly close to $1/2$ as well. ■

4 Chosen-Ciphertext Secure BTE Schemes

The techniques of the previous section may also be used to construct a BTE scheme secure in the sense of $\mathsf{SN\text{-}CCA}$ from any BTE scheme secure in the sense of $\mathsf{SN\text{-}CPA}$. Roughly, we view the subtree of each node as a (hierarchical) IBE scheme, and use the scheme from the previous section for that subtree. We first give a high-level overview for the simpler case of a BTE scheme which only allows encryption to nodes at a single depth ℓ (as opposed to a full-fledged BTE scheme which allows encryption to nodes at *all* depths $\leq \ell$). To encrypt a message for node w, the sender generates keys (vk, sk) for a one-time signature scheme (as in the previous section) and encrypts the message m for "node" $w|vk$ to obtain ciphertext C; the sender additionally signs C using sk resulting in signature σ. The complete ciphertext is $\langle vk, C, \sigma \rangle$. When node w, holding secret key SK_w, receives a ciphertext of this form, it first verifies that the signature is correct with respect to vk. If so, the receiver computes secret key $SK_{w|vk}$ on its own (using repeated applications of the Der algorithm) and then uses this key to recover m from C. As for the scheme from the previous section, the intuition here is that encryption to "node" $w|vk$ is secure even if an adversary can obtain secret keys for "nodes" $w'|vk'$ (with $(w', vk') \neq (w, vk)$). Thus, even more so, encryption to "node" $w|vk$ remains secure if the adversary can obtain (only) *decryptions* of ciphertexts intended for "nodes" $w'|vk'$ of this sort. And of course, the adversary is unable to obtain any decryptions for "node" $w|vk$ itself unless it can forge a new signature with respect to vk.

The construction is a bit more involved for the case of general BTE (i.e., when encryption is allowed to nodes at arbitrary depth rather than at a single depth). The issue that we must resolve is the encoding of node names (for example, we must ensure $w|vk$ is not mapped to the same node as some other w'). A simple way of resolving this issue is to encode each node name $w = w_1 w_2 \ldots w_t$ as $1w_1 1w_2 \ldots 1w_t$, and then encode $w|vk$ as $1w_1 1w_2 \ldots 1w_t 0|vk$. We describe the full construction in detail below.

Let $\Pi' = (\mathsf{Setup}', \mathsf{Der}', \mathcal{E}', \mathcal{D}')$ be a BTE scheme and let $\mathsf{Sig} = (\mathcal{G}, \mathsf{Sign}, \mathsf{Vrfy})$ be a one-time signature scheme in which the verification key output by $\mathcal{G}(1^k)$ has length $\ell_s(k)$. As in the previous section, we require this scheme to be secure in the sense of *strong* unforgeability. Next, define a function Encode on strings w such that:

$$\mathsf{Encode}(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ 1w_1 1w_2 \cdots 1w_t & \text{if } w = w_1 \cdots w_t \text{ with } w_i \in \{0,1\}. \end{cases}$$

(Note that $|\mathsf{Encode}(w)| = 2|w|$.) The construction of BTE scheme $\Pi = (\mathsf{Setup}, \mathsf{Der}, \mathcal{E}, \mathcal{D})$ proceeds as follows:

- $\mathsf{Setup}(1^k, \ell)$ runs $\mathsf{Setup}'(1^k, 2\ell + \ell_s(k) + 1)$ to obtain (PK, SK_ε). The system-wide public key is PK and the root secret key is SK_ε.

- $\mathsf{Der}(w, SK_w)$ proceeds as follows. First, set $w' = \mathsf{Encode}(w)$. Next, compute $SK'_{w'1}$ using $\mathsf{Der}'_{SK_w}(w')$ followed by $(SK_{w'10}, SK_{w'11}) \leftarrow \mathsf{Der}_{SK'_{w'1}}(w'1)$. Set $SK_{w0} = SK'_{w'10}$ and $SK_{w1} = SK'_{w'11}$ and output (SK_{w0}, SK_{w1}). (Note that $w'10 = \mathsf{Encode}(w0)$ and analogously for $w'11$.)

  Intuitively, any node w in scheme Π is mapped to a node $w' = \mathsf{Encode}(w)$ in Π'. Thus, secret key SK_w for node w (in Π) corresponds to secret key $SK'_{w'}$ for node w' (in Π'). So, to derive the secret keys for the children of w (i.e., $w0, w1$) in Π, we must derive the keys for the (right) *grandchildren* of node w' in Π'.

- To encrypt message m for a particular node $w \in \{0,1\}^{\leq \ell}$ using public parameters PK, the sender first runs $\mathcal{G}(1^k)$ to obtain verification key vk and signing key sk. Next, the sender sets $w' = \mathsf{Encode}(w)$. The sender then computes $C \leftarrow \mathcal{E}'_{PK}(w'|0|vk, m)$ (i.e., the sender encrypts m with respect to "node" $w'|0|vk$ using Π') and $\sigma \leftarrow \mathsf{Sign}_{sk}(C)$. The final ciphertext is $\langle vk, C, \sigma \rangle$.

- Node w, with secret key SK_w, decrypts a ciphertext $\langle vk, C, \sigma \rangle$ as follows. First, check whether $\mathsf{Vrfy}_{vk}(C, \sigma) \stackrel{?}{=} 1$. If not, simply output $\perp$. Otherwise, let $w' = \mathsf{Encode}(w)$. The receiver then computes the secret key $SK'_{w'|0|vk}$ using repeated applications of Der', and outputs $m \leftarrow \mathcal{D}'_{SK'_{w'|0|vk}}(w'|0|vk, C)$.

We now state the main result of this section:

Theorem 2 *If Π' is a* BTE *scheme which is secure in the sense of* SN-CPA *and* Sig *is a strongly unforgeable one-time signature scheme, then Π is a* BTE *scheme which is secure in the sense of* SN-CCA.

Proof The proof is largely similar to that of Theorem 1. Given any PPT adversary $\mathcal{A}$ attacking Π in a selective node, (adaptive) chosen ciphertext attack, we construct a PPT adversary $\mathcal{A}'$ attacking Π' in a selective node, chosen-plaintext attack. Relating the success probabilities of these adversaries gives the desired result.

We first define an event Forge; because we are working in the context of BTE, the definition is slightly different from the definition used in the proof of Theorem 1. Specifically, let w^* denote the node initially output by $\mathcal{A}$, and let $\langle vk^*, C^*, \sigma^* \rangle$ be the challenge ciphertext received by $\mathcal{A}$. Now, let Forge denote the event that $\mathcal{A}$ makes a decryption query $\widehat{\mathcal{D}}(w^*, \langle vk^*, C', \sigma' \rangle)$ with $(C', \sigma') \neq (C^*, \sigma^*)$ but for which $\mathsf{Vrfy}_{vk^*}(C', \sigma') = 1$. (We include in this event the case when $\mathcal{A}$ submits such a query to its decryption oracle *before* receiving the challenge ciphertext; in this case, we do not require $(C', \sigma') \neq (C^*, \sigma^*)$.) It is easy to see that we can use $\mathcal{A}$ to break the underlying one-time signature scheme Sig with probability exactly $\Pr_{\mathcal{A}}[\mathsf{Forge}]$; since Sig is a strongly unforgeable one-time signature scheme, it must be the case that $\Pr_{\mathcal{A}}[\mathsf{Forge}]$ is negligible (in the security parameter k).

We now define adversary $\mathcal{A}'$ as follows:

1. $\mathcal{A}'(1^k, \ell')$ sets $\ell = (\ell' - \ell_s(k) - 1)/2$ and runs $\mathcal{A}(1^k, \ell)$ who, in turn, outputs a node $w^* \in \{0,1\}^{\leq \ell}$. Adversary $\mathcal{A}'$ sets $w' = \mathsf{Encode}(w^*)$, and runs $\mathcal{G}(1^k)$ to generate (vk^*, sk^*). Finally, $\mathcal{A}'$ outputs the node $w^{*\prime} = w'|0|vk^*$.

2. $\mathcal{A}'$ is given PK as well as a set of secret keys $\{SK'_w\}$ for all nodes w of the following form:

   – $w = v\bar{b}$, where vb is a prefix of $w^{*\prime}$ and $b \in \{0,1\}$;

   – $w = w^{*\prime}0$ or $w = w^{*\prime}1$ (in case $|w^{*\prime}| < \ell'$).

   Using these, $\mathcal{A}'$ can compute and give to $\mathcal{A}$ all the relevant secret keys that $\mathcal{A}$ expects.

3. When $\mathcal{A}$ makes decryption query $\widehat{\mathcal{D}}(w, \langle vk, C, \sigma \rangle)$, adversary $\mathcal{A}'$ proceeds as follows:

   (a) If $\mathsf{Vrfy}_{vk}(C, \sigma) \neq 1$, then $\mathcal{A}'$ simply returns $\perp$.

   (b) If $w = w'$, $\mathsf{Vrfy}_{vk}(C, \sigma) = 1$, and $vk = vk^*$ (i.e., event Forge occurs), then $\mathcal{A}'$ halts and outputs a random bit.

   (c) Otherwise, set $\tilde{w} = \mathsf{Encode}(w)$. Note that $\mathcal{A}'$ is able to derive the secret key corresponding to the "node" $\tilde{w}|0|vk$ using the secret keys it obtained in step 2 (this follows since $\tilde{w}|0|vk$ cannot be a prefix of $w^{*\prime}$). So, $\mathcal{A}'$ simply computes the necessary key, performs the decryption of C, and returns the result to $\mathcal{A}$.

4. When $\mathcal{A}$ outputs its two messages m_0, m_1, these same messages are output by $\mathcal{A}'$. In return, $\mathcal{A}'$ receives a ciphertext C^*. Adversary $\mathcal{A}'$ computes $\sigma^* \leftarrow \mathsf{Sign}_{sk^*}(C^*)$ and returns ciphertext $\langle vk^*, C^*, \sigma^* \rangle$ to $\mathcal{A}$.

5. Any subsequent decryption queries of $\mathcal{A}$ are answered as before.

6. Finally, $\mathcal{A}$ outputs a guess b'; this same guess is output by $\mathcal{A}'$.

Note that $\mathcal{A}'$ represents a legal adversarial strategy for attacking Π'. Furthermore, $\mathcal{A}'$ provides a perfect simulation for $\mathcal{A}$ (and thus $\mathcal{A}'$ succeeds whenever $\mathcal{A}$ succeeds) unless event Forge occurs. An analysis as in the proof of Theorem 1 shows that $\Pr_{\mathcal{A},\Pi}[\mathsf{Succ}]$ must be negligibly close to 1/2. ■

Applications to FSE and HIBE. Canetti, et al. [5] have shown that any BTE scheme can be used to construct both a forward-secure public-key encryption scheme as well as a "full-fledged" hierarchical ID-based encryption scheme (and, as a special case, an ID-based encryption scheme). Furthermore, if the original BTE scheme is secure against (adaptive) chosen-ciphertext attacks, then so is the derived scheme. In their original work [5], Canetti, et al. suggested that a BTE scheme secure in the sense of SN-CCA could be derived using the Naor-Yung paradigm [18] along with 1-time, simulation-sound NIZK proofs [20]. As mentioned in the Introduction, the use of NIZK proofs results in a completely impractical scheme (at least using currently-known techniques). Thus, our approach of this section provides a more efficient way of achieving CCA security for any BTE scheme (as well as CCA security for forward-secure encryption or HIBE) in the standard model.

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. *Adv. in Cryptology — Crypto 1998*, LNCS vol. 1462, Springer-Verlag, pp. 26–45, 1998.

[2] M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. *First ACM Conf. on Computer and Communications Security*, ACM, pp. 62–73, 1993.

[3] M. Blum, P. Feldman, and S. Micali. Non-Interactive Zero-Knowledge and its Applications. *20th ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 103–112, 1988.

[4] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. Crypto 2001. Full version to appear in *SIAM J. Computing* and available at http://eprint.iacr.org/2001/090.

[5] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. *Adv. in Cryptology — Eurocrypt 2003*, LNCS vol. 2656, Springer-Verlag, pp. 255–271, 2003. Full version available at http://eprint.iacr.org/2003/083.

[6] C. Cocks. An Identity-Based Encryption Scheme Based on Quadratic Residues. *Cryptography and Coding*, 2001.

[7] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Chosen Ciphertext Attack. *Adv. in Cryptology — Crypto 1998*, LNCS vol. 1462, Springer-Verlag, pp. 13–25, 1998.

[8] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. *Adv. in Cryptology — Eurocrypt 2002*, LNCS vol. 2332, Springer-Verlag, pp. 45–64, 2002.

[9] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM J. Computing* 30(2): 391–437, 2000.

[10] E. Elkind and A. Sahai. A Unified Methodology For Constructing Public-Key Encryption Schemes Secure Against Adaptive Chosen-Ciphertext Attack. Available from http://eprint.iacr.org/2002/042/.

[11] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Under General Assumptions. *SIAM J. Computing* 29(1): 1–28, 1999. A preliminary version appeared in FOCS '90.

[12] C. Gentry and A. Silverberg. Hierarchical Identity-Based Cryptography. *Adv. in Cryptology — Asiacrypt 2002*, LNCS vol. 2501, Springer-Verlag, pp. 548–566, 2002.

[13] Y. Gertner, T. Malkin, and S. Myers. Personal Communication.

[14] S. Goldwasser and S. Micali, Probabilistic encryption, *JCSS,* Vol. 28, No 2, 1984.

[15] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption. *Adv. in Cryptology — Eurocrypt 2002*, LNCS vol. 2332, Springer-Verlag, pp. 466–481, 2002.

[16] L. Lamport. Constructing Digital Signatures from a One-Way Function. Technical Report CSL-98, SRI International, Palo Alto, 1979.

[17] Y. Lindell. A Simpler Construction of CCA-Secure Public-Key Encryption Under General Assumptions. *Adv. in Cryptology — Eurocrypt 2003*, LNCS vol. 2656, Springer-Verlag, pp. 241–254, 2003.

[18] M. Naor and M. Yung. Public-Key Cryptosystems Provably-Secure against Chosen-Ciphertext Attacks. *22nd ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 427–437, 1990.

[19] C. Rackoff and D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *Adv. in Cryptology — Crypto 1991*, LNCS vol. 576, Springer-Verlag, pp. 433–444, 1991.

[20] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. *40th IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 543–553, 1999.

[21] A. Shamir. Identity-based cryptosystems and signature schemes. *Adv. in Cryptology — Crypto 1984*, LNCS vol. 196, Springer-Verlag, pp. 47–53, 1984.

[22] V. Shoup. Why Chosen Ciphertext Security Matters. IBM Research Report RZ 3076, November, 1998. Available at http://www.shoup.net/papers.

[23] V. Shoup. A Proposal for an ISO Standard for Public Key Encryption (version 2.1). December, 2001. Available at http://www.shoup.net/papers.