Chosen-Ciphertext Security from Identity-Based Encryption

Ran Canetti[1], Shai Halevi[1], and Jonathan Katz[2*]

[1] IBM T. J. Watson Research Center, Hawthorne, NY.
{canetti,shaih}@watson.ibm.com
[2] Dept. of Computer Science, University of Maryland, College Park, MD.
jkatz@cs.umd.edu

Abstract. We propose a simple and efficient construction of a CCA-secure public-key encryption scheme from any CPA-secure identity-based encryption (IBE) scheme. Our construction requires the underlying IBE scheme to satisfy only a relatively "weak" notion of security which is known to be achievable without random oracles; thus, our results provide a new approach for constructing CCA-secure encryption schemes in the standard model. Our approach is quite different from existing ones; in particular, it avoids non-interactive proofs of "well-formedness" which were shown to underlie most previous constructions. Furthermore, applying our conversion to some recently-proposed IBE schemes results in CCA-secure schemes whose efficiency makes them quite practical.

Our technique extends to give a simple and reasonably efficient method for securing any binary tree encryption (BTE) scheme against adaptive chosen-ciphertext attacks. This, in turn, yields more efficient CCA-secure hierarchical identity-based and forward-secure encryption schemes in the standard model.

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

1 Introduction

Security against adaptive chosen-ciphertext attacks (i.e., "CCA security") is a strong and very useful notion of security for public-key encryption schemes [RS91,DDN00,BDPR98] . 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., [S98].

However, there are only a handful of known public-key encryption schemes that can be proven 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

* Work supported by NSF Trusted Computing Grant #ANI-0310751.

proposed for constructing such cryptosystems. The first follows the paradigm of Naor and Yung [NY90] (later extended by Sahai [S99] and simplified by Lindell [L03]), and the related scheme of Dolev, Dwork, and Naor [DDN00]. This technique uses as building blocks any CPA-secure public-key encryption scheme (i.e., any scheme secure against chosen-plaintext attacks [GM84]) along with any non-interactive zero-knowledge (NIZK) proof system [BFM88,FLS90]; in turn, each of these primitives may be constructed using any family of trapdoor permutations. The encryption schemes resulting from this approach, however, are highly inefficient precisely because they employ NIZK proofs which in turn use a generic Karp reduction from an instance of the encryption scheme to an instance of some NP-complete problem. Furthermore, there are currently no known efficient NIZK proof systems even under specific assumptions and for particular cryptosystems of interest. Thus, given current techniques, this methodology for constructing CCA-secure cryptosystems serves as a "proof of feasibility" but does not lead to practical constructions.

The second technique is due to Cramer and Shoup [CS98,CS02], and is based on algebraic constructs with particular homomorphic properties (namely, those which admit "smooth hash proof systems"; see [CS02]). Algebraic constructs of the appropriate type are known to exist based on some specific assumptions, namely the hardness of the decisional Diffie-Hellman problem [CS98] or the hardness of deciding quadratic residuosity or N^{th} residuosity in certain groups [CS02]. More efficient schemes following the same basic technique have been given recently [GL03,CS03], and the technique leads to a number of possible instantiations which are efficient enough to be used in practice.

Interestingly, as observed by Elkind and Sahai [ES02], 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, and thus all the schemes mentioned above (with the possible exception of [DDN00]) may be viewed as instantiations of a single paradigm.

1.1 Our contributions

We propose a new approach for constructing CCA-secure public-key encryption schemes. Instead of using "proofs of well-formedness" as in all previous schemes, we instead give a direct construction using identity-based encryption (IBE) schemes satisfying a "weak" notion of security. A number of IBE schemes meeting this weak notion of security in the standard model were recently proposed (see below); thus, our approach yields new constructions of CCA-secure encryption in the standard model. The resulting schemes are simple and reasonably efficient, and are quite different from the ones described above. In particular, they do not seem to fit within the characterization of Elkind and Sahai. We remark that our techniques may also be used to construct a non-adaptive (or "lunchtime") CCA1-secure encryption scheme [NY90,DDN00,BDPR98] based on any weak IBE scheme; interestingly, our conversion in this case adds (essentially) *no overhead* to the original IBE scheme.

Before sketching our construction, we first recall the notion of IBE. The concept of identity-based encryption was introduced by Shamir [S84], and provably-secure IBE schemes (in the random oracle model) were recently demonstrated by Boneh and Franklin [BF01] and Cocks [C01]. 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 [BF01], 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, proposed by Canetti, Halevi, and Katz [CHK03], 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. Canetti, et al. [CHK03] show that a weak IBE scheme can be constructed based on the BDH assumption *in the standard model*. Concurrent with the present work, more efficient constructions of weak IBE schemes in the standard model (including one based on the BDH assumption) were given by Boneh and Boyen [BB04]. Both of the above-mentioned constructions of weak IBE based on the BDH assumption build on earlier work of Gentry and Silverberg [GS02].

Our construction of CCA-secure encryption requires only an IBE scheme satisfying the weaker notion of security referred to above. The conversion of any such IBE scheme to a CCA-secure 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 a one-time strong signature scheme, and then encrypts the message with respect to the "identity" vk. (A "strong" signature scheme has the property that it is infeasible to create new valid signature even for previously-signed messages.) 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 a ciphertext $\langle vk, C, \sigma \rangle$, the receiver first verifies the signature on C

with respect to vk, and outputs $\perp$ if the verification fails. Otherwise, the receiver 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 (strong) security of the one-time signature scheme. The crux of the security proof then involves 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 encrypted for "identity" vk^* using an IBE scheme).

A simple modification of the above gives a (non-adaptive) CCA1-secure scheme with virtually no overhead compared to the original IBE scheme. Namely, replace the verification key vk by a randomly-chosen string $r \in \{0,1\}^k$ (and forego any signature); the resulting ciphertext is simply $\langle r, C \rangle$, where C is encrypted with respect to the "identity" r. Since an adversary cannot guess in advance which r a sender will use, an argument similar to the above shows that this scheme is secure against non-adaptive chosen-ciphertext attacks.

Straightforward implementation of the above ideas using the "weak IBE" construction from [CHK03] is still rather inefficient; in particular, decryption requires computation of (roughly) one bilinear mapping per bit of the verification key. (Using standard hashing techniques, however, one can obtain a signature scheme in which the length of the verification key is exactly the security parameter.) One can somewhat optimize this construction by working with trees of high degree instead of binary trees as in [CHK03]. Specifically, using a tree of degree d results in a scheme requiring $n/\log_2 d$ mapping computations for an n-bit verification key; in this case we pay for these savings by having to increase the key size by a factor of d. (We speculate that using $d = 16$ results in a "borderline practical" scheme.) Alternatively, using one of the weak IBE schemes proposed by [BB04] results in a considerably more efficient scheme, including one which is nearly as efficient as the Cramer-Shoup cryptosystem [CS98].

Further extensions and applications. Canetti, Halevi, and Katz [CHK03] propose the notion of binary tree encryption (BTE), show how to construct a secure BTE scheme in the standard model, and furthermore show how to construct both hierarchical IBE (HIBE) schemes [HL02,GS02] and forward-secure encryption (FSE) schemes starting from any BTE scheme, again in the standard model. To obtain security against chosen-ciphertext attacks in each of these cases, they suggest using the technique of Naor and Yung [NY90] as adapted by Sahai and Lindell [S99,L03]. 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 CPA-secure BTE scheme to a CCA-secure BTE scheme. The resulting BTE scheme is considerably more efficient than a 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. Our techniques may also be used *directly* to convert any CPA-secure HIBE scheme to a CCA-secure HIBE scheme, with possibly improved efficiency.

Implications for "black-box" separations. Our construction of a CCA-secure encryption scheme from any weak 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 scheme). A recent result of Aiello, et al. [AGMM04] rules out certain classes of black-box constructions of CCA-secure encryption schemes from CPA-secure ones. Combined with their result, the current work rules out the same classes of black-box constructions of IBE from CPA-secure encryption.

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 [DDN00,S99] rely on NIZK and are inherently *non*-black box), it does rule out certain techniques for constructing IBE schemes based on general assumptions.

Related work. In recent and independent work, MacKenzie, Reiter, and Yang [MRY04] introduce the notion of tag-based non-malleability (tnm), give efficient constructions of "tnm-cca-secure" cryptosystems in the random oracle model, and show how to construct a CCA-secure cryptosystem from any tnm-cca-secure scheme. Interestingly, their conversion from tnm-cca security to (full) CCA security uses a one-time signature scheme in essentially the same way that we do. Viewed in the context of their results, our results of Section 3 give an efficient construction of a tnm-cca-secure scheme from any weak IBE scheme, and imply an efficient and novel construction of a tnm-cca-secure scheme *in the standard model*. Our results of Section 4 have no counterpart in [MRY04].

2 Definitions

2.1 Public-Key Encryption

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

- *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 against adaptive chosen-ciphertext attacks (cf. [BDPR98]).

Definition 2. *A public-key encryption scheme* PKE *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 that 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 definitions of Boneh and Franklin [BF01].

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 $m \in \{0,1\}^*$; it 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. [CHK03]. 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. An IBE scheme satisfying this definition suffices for our purposes. Furthermore, schemes satisfying this definition of security in the standard model are known [CHK03,BB04]. (For the case of the original definition of Boneh and Franklin, only constructions in the random oracle model are known.)

Definition 4. *An identity-based scheme* IBE *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 that 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* $|\Pr_{A,\mathsf{IBE}}[\mathsf{Succ}] - 1/2|$.

The above definition may be extended in the obvious way to encompass security against (adaptive) chosen-ciphertext attacks. In this case, 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 $\mathsf{Der}_{\mathsf{msk}}(ID^*)$).[3] 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 [CHK03], and may be viewed as a relaxed variant of hierarchical identity-based encryption (HIBE) [HL02,GS02] in the following sense: in a BTE scheme, each node has two children (labeled "0" and "1") while in a HIBE scheme, each node

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

has arbitrarily-many children labeled with arbitrary strings. Although BTE is seemingly weaker than HIBE, it is known [CHK03] that a BTE scheme supporting a 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). We review the relevant definitions of Canetti, et al. [CHK03].

Definition 5. *A binary tree encryption scheme* BTE *is a 4-tuple of* PPT *algorithms* (Setup, 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 [CHK03], is weaker than the notion of security originally considered by Gentry and Silverberg [GS02]. As in the definition of security for ID-based encryption given in the previous section, 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, and a construction meeting this definition of security in the standard model is known [CHK03]. (In contrast, a construction meeting the stronger security definition of [GS02] is known only in the random oracle model and only for trees of constant depth).

Definition 6. *A binary tree encryption scheme* BTE *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* P *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 assumes* $|w^*| < \ell$*).*

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". The above definition may also be extended in the natural way to encompass security against (adaptive) chosen-ciphertext attacks. (We refer to schemes meeting this definition of security as "secure in the sense of SN-CCA".) Such a definition can be found in [CHK03], 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 require that this scheme be secure in the sense of *strong* unforgeability (i.e., an adversary is unable to forge even a new signature on a previously-signed message). We note that such a scheme may be based on any one-way function [L79,R90] 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 $\mathsf{Vrfy}_{vk}(C, \sigma) \stackrel{?}{=} 1$. 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 attacks.*

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 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, $\mathcal{A}$ may 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), and since $\Pr_{\mathcal{A}}[\mathsf{Forge}]$ is negligible, it must be the case that $\Pr_{\mathcal{A},\Pi}[\mathsf{Succ}]$ is negligibly close to $1/2$ as well.

4 Chosen-Ciphertext Security for BTE Schemes

The techniques of the previous section may also be used to construct a BTE scheme secure in the sense of SN-CCA from any BTE scheme secure in the sense of SN-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 multiple "nodes" $w'|vk'$ with $(w', vk') \neq (w, vk)$ (recall we are assuming here that all nodes w are at the same depth, so $w'|vk'$ cannot be a prefix of $w|vk$).

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 schemes (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 that $w|vk$ is not mapped to the same node as some other w'. A simple way of resolving this issue is to encode each node name $w = w_1 w_2 \dots w_t$ as $1w_1 1w_2 \dots 1w_t$, and then encode $w|vk$ as $1w_1 1w_2 \dots 1w_t 0|vk$. 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 binary tree encryption 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 corresponds 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 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 $m \leftarrow \mathcal{D}'_{SK'_{w'|0|vk}}(w'|0|vk, C)$.

Remark 1. The above approach can be used to derive a CCA-secure HIBE scheme from a CPA-secure HIBE scheme in the following way: CPA-secure HIBE trivially implies CPA-secure BTE; the conversion above yields CCA-secure BTE; and the latter implies CCA-secure HIBE (see [CHK03]). However, it will in general be much more efficient to apply the above techniques *directly*: In this case, we would simply encode the ID-vector $\boldsymbol{w} = w_1|\cdots|w_t$ as $\boldsymbol{w}' = 1w_1|\cdots|1w_t$, and encode $\boldsymbol{w}|vk$ as an ID-vector $\boldsymbol{w}'|0vk$.

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, 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 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.

The above construction requires only a one-time signature scheme in addition to the underlying BTE scheme; the existence of the former (which may be constructed from any one-way function) is implied by the existence of any BTE scheme secure in the sense of SN-CPA. Putting these observations together shows:

Theorem 3. *If there exists a BTE scheme secure in the sense of* SN-CPA*, then there exists a BTE scheme secure in the sense of* SN-CCA*.*

Note that an analogous result for the case of (standard) public-key encryption is not known.

Further applications. In [CHK03] it is shown that any BTE scheme can be used to construct both a forward-secure public-key encryption scheme as well as a "full-fledged" HIBE scheme (and, as a special case, an IBE scheme). Furthermore, if the original BTE scheme is secure against chosen-ciphertext attacks, then so are the derived schemes. Canetti, et al. further suggest [CHK03] that a BTE scheme secure in the sense of SN-CCA can be derived using the Naor-Yung paradigm [NY90] along with 1-time, simulation-sound NIZK proofs [S99]. As mentioned in the Introduction, the use of NIZK proofs results in a completely impractical scheme, at least using currently-known techniques. Thus, the 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. (See also Remark 1.)

When our techniques are applied to a BTE/IBE/HIBE scheme secure against selective-node/identity attacks, the resulting CCA-secure scheme is also only resilient to selective-node/identity attacks. However, when our techniques are applied to stronger schemes which are CPA-secure against an *adaptive* choice of node/identity, the resulting CCA-secure scheme maintains this level of security as well.

We remark that when the transformation outlined in this section is applied to the recent constructions of Boneh and Boyen [BB04], we obtain truly practical constructions of IBE and HIBE schemes secure against selective-identity, chosen-ciphertext attacks in the standard model.

Acknowledgments

We thank Eu-Jin Goh for pointing out that our techniques imply a conversion from weak IBE to "lunchtime" CCA1 security with essentially no overhead.

References

[AGMM04] B. Aiello, Y. Gertner, T. Malkin, and S. Myers. Personal communication.

[BDPR98] 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.

[BFM88] 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.

[BB04] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. *Adv. in Cryptology — Eurocrypt 2004*, to appear.

[BF01] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *Adv. in Cryptology — Crypto 2001*, LNCS vol. 2139, Springer-Verlag, pp. 213–229, 2001. Full version to appear in *SIAM J. Computing* and available at `http://eprint.iacr.org/2001/090`.

[CHK03] 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`.

[C01] C. Cocks. An Identity-Based Encryption Scheme Based on Quadratic Residues. *Cryptography and Coding*, LNCS vol. 2260, Springer-Verlag, pp. 360–363, 2001.

[CS98] 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.

[CS02] 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.

[CS03] J. Camenisch and V. Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms. *Adv. in Cryptology — Crypto 2003*, LNCS vol. 2729, Springer-Verlag, pp. 126–144, 2003.

[DDN00] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM J. Computing* 30(2): 391–437, 2000.

[ES02] E. Elkind and A. Sahai. A Unified Methodology For Constructing Public-Key Encryption Schemes Secure Against Adaptive Chosen-Ciphertext Attack. *First Theory of Cryptography Conference (TCC)* 2004, to appear. Available from `http://eprint.iacr.org/2002/042/`.

[FLS90] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Under General Assumptions. *SIAM J. Computing* 29(1): 1–28, 1999.

[GL03] R. Gennaro and Y. Lindell. A Framework for Password-Based Authenticated Key Exchange. *Adv. in Cryptology — Eurocrypt 2003*, LNCS vol. 2656, Springer-Verlag, pp. 524–543, 2003.

[GS02] C. Gentry and A. Silverberg. Hierarchical Identity-Based Cryptography. *Adv. in Cryptology — Asiacrypt 2002*, LNCS vol. 2501, Springer-Verlag, pp. 548–566, 2002.

[GM84] S. Goldwasser and S. Micali. Probabilistic Encryption. *J. Computer System Sciences* 28(2): 270–299, 1984.

[HL02] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption. *Adv. in Cryptology — Eurocrypt 2002*, LNCS vol. 2332, Springer-Verlag, pp. 466–481, 2002.

[L79] L. Lamport. Constructing Digital Signatures from a One-Way Function. Technical Report CSL-98, SRI International, Palo Alto, 1979.

[L03] 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.

[MRY04] P. MacKenzie, M. Reiter, and K. Yang. Alternatives to Non-Malleability: Definitions, Constructions, and Applications. *First Theory of Cryptography Conference (TCC)* 2004, to appear.

[NY90] 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.

[RS91] 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, 1992.

[R90] J. Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. *22nd ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 387–394, 1990.

[S99] 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.

[S84] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. *Adv. in Cryptology — Crypto 1984*, LNCS vol. 196, Springer-Verlag, pp. 47–53, 1985.

[S98] V. Shoup. Why Chosen Ciphertext Security Matters. IBM Research Report RZ 3076, November, 1998. Available at `http://www.shoup.net/papers`.