Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption

Dan Boneh*   Jonathan Katz†

Abstract

Recently, Canetti, Halevi, and Katz showed a general method for constructing CCA-secure encryption schemes from identity-based encryption schemes in the standard model. We improve the efficiency of their construction, and show two specific instantiations of our resulting scheme which offer the most efficient encryption (and, in one case, key generation) of any CCA-secure encryption scheme to date.

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

1 Introduction

Security against adaptive chosen-ciphertext attacks (i.e., "CCA-security") [29, 17, 1] has become the *de facto* level of security for public-key encryption schemes. The reasons for this are many: CCA security helps protect against subtle attacks that have been demonstrated against schemes *not* meeting this notion of security [3, 24, 23]; is helpful in defending against "active" attackers who may modify messages in transit (see [32]); and, finally, allows encryption schemes to be developed and then securely "plugged in" to higher-level protocols which may then be executed in arbitrary environments (see, e.g., [8, Sec. 8.2.2]).

Nevertheless, only a relatively small number of encryption schemes have been rigorously proven secure against adaptive chosen-ciphertext attacks *in the standard model*[1] (i.e., without resorting to the use of random oracles [2]). Schemes based on general assumptions are known [17, 30, 27], but these rely on generic non-interactive zero-knowledge proofs [4, 18] and do not currently lead to practical solutions. More interesting from a practical point of view are efficient schemes based on specific number-theoretic assumptions; two general methodologies for constructing such schemes are known. The first methodology is based on the "smooth hash proof systems" of Cramer and Shoup [14], and has led to a variety of constructions [13, 14, 19, 15, 25]. The second, and more recent, method [11] constructs a CCA-secure encryption scheme from any semantically-secure (or, "CPA-secure") identity-based encryption (IBE) scheme [7, 12] (which can in turn be constructed in the standard model based on specific number-theoretic assumptions [10, 5, 6, 34]). Overall, the most efficient CCA-secure encryption scheme currently known is a hybrid encryption system due to Kurosawa and Desmedt [25] which builds on the original proposal of Cramer and Shoup [13] and relies on the decisional Diffie-Hellman assumption.

*`dabo@cs.stanford.edu`. Computer Science Department, Stanford University. Supported by NSF and the Packard Foundation.

†`jkatz@cs.umd.edu`. Department of Computer Science, University of Maryland. This research was supported by NSF Trusted Computing Grant #0310751.

[1] From now on, we use "CCA security" to refer by default to security which is proven in the standard model.

In this paper, we suggest a new method which allows for the construction of very efficient CCA-secure encryption schemes. Our technique modifies the approach of Canetti, Halevi, and Katz [11], who (as noted above) show a transformation from any semantically-secure "weak" IBE scheme to a CCA-secure public-key encryption scheme. Briefly and somewhat informally, their transformation from an IBE scheme[2] $(\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ to a CCA-secure scheme proceeds as follows: key generation is performed by running Setup and letting the public (resp. secret) key be the master public key PK (resp., master secret key msk) output by this algorithm. To encrypt a message m using public key PK, a sender generates a random key-pair (vk, sk) for a one-time signature scheme and sends the ciphertext $\langle vk, \mathsf{Enc}_{PK}(vk, m), \sigma\rangle$, where $\mathsf{Enc}_{PK}(vk, m)$ represents an encryption of message m for the "identity" vk using master public parameters PK, and σ represents a signature on the second component of this ciphertext using sk. To decrypt ciphertext $\langle vk, C, \sigma\rangle$, the receiver first verifies whether $\mathsf{Vrfy}_{vk}(C, \sigma) \stackrel{?}{=} 1$. If so, the receiver then decrypts C with respect to the "identity" vk (it can do this since it has the master secret key msk).

Though conceptually simple, this transformation does add noticeable overhead to the underlying IBE scheme: encryption requires the sender to generate keys for a one-time signature scheme [26] and also to compute a signature using the keys just generated; decryption requires the receiver to verify a signature with respect to the verification key included as part of the ciphertext. Although one-time signatures are "easy" to construct in theory, and are more efficient than "full-blown" signatures (i.e., those which are existentially unforgeable under an adaptive chosen-message attack [20]), they still have their price. In particular:

- One-time signatures based on cryptographic hash functions such as SHA-1 can be designed to allow very efficient *signing*; key generation, on the other hand, typically requires hundreds of hash function evaluations and is relatively expensive (though not as expensive as key generation in schemes based on number-theoretic assumptions). More problematic, perhaps, is that such schemes have very long public keys and signatures, which would result in very long ciphertexts in the scheme of [11].

- One-time signatures based on number-theoretic assumptions (say, by adapting "full-blown" signature schemes) yield schemes whose computational cost — both for key generation and signing — is more expensive, but which have the advantage of short(er) public keys and signatures.

Either way, the transformation of Canetti, Halevi, and Katz results in a CCA-secure encryption scheme which is less efficient than the underlying IBE system.

1.1 Our Contribution

We describe a transformation from any CPA-secure "weak" IBE system to a CCA-secure encryption scheme which adds essentially no overhead. The efficiency advantage of our approach arises from our observation that the one-time signature in the construction of Canetti, et al. (as described earlier) can be replaced by a message-authentication code (MAC) along with an appropriate "encapsulation" of a MAC key (for the purposes of this informal description, one can think of an encapsulation as a commitment). Using the notation introduced earlier, encryption using our approach is now performed (informally) by first "encapsulating" a key r which results in an encapsulation com along with a decommitment string dec. The final ciphertext is $\langle \mathsf{com}, \mathsf{Enc}_{PK}(\mathsf{com}, m \circ \mathsf{dec}), \mathsf{tag}\rangle$, where tag

[2] Definitions of IBE schemes and their security, as well as definitions of CCA-secure encryption, are reviewed in Section 2.

is now a message authentication code computed on the second component of the ciphertext using key r. Decryption of ciphertext $\langle \mathsf{com}, C, \mathsf{tag} \rangle$ is done in the natural way, but note that here the receiver must first decrypt C (with respect to "identity" com) and only then can the receiver verify the correctness of tag. Indeed, this feature of our scheme complicates the security proof somewhat (and in particular we must be careful to avoid circular arguments).

Adapting [16, 21], we show how encapsulation of the MAC key can be done both efficiently and securely using, e.g., SHA-1: encapsulation requires only a single hash function evaluation, and is secure under the assumption that SHA-1 is second-preimage resistant (the scheme can be easily modified so as to be secure under the weaker assumption of the existence of UOWHFs [28]). This encapsulation scheme may have other applications, and thus the scheme — as well as the relatively simple proof of security we provide for this encapsulation scheme here (cf. Theorem 2) — may be of independent interest. Furthermore, our technique of replacing a one-time signature by a MAC seems applicable to other constructions (e.g., those of [17, 30] as well as the various extensions mentioned in [11]), giving efficiency improvements in those cases as well.

In addition to the general method discussed above, we also show two specific instantiations of our approach based on two IBE schemes recently introduced by Boneh and Boyen [5]. Our resulting schemes are quite efficient: in particular, the times required for key generation and encryption are as fast as (or faster than) the most efficient previous CCA-secure schemes to date.

1.2 Hybrid Encryption

In practice, public-key encryption is almost never used to encrypt actual data. Instead, *hybrid encryption* is typically used, whereby a public-key scheme is used to encrypt a random key, and the data is then encrypted using some symmetric-key encryption scheme and this key. In fact, "encryption" of the symmetric key is not required; "encapsulation" (cf. [33]) — which may be more efficient — is enough. It is well known that if both the public-key encapsulation scheme and the underlying symmetric-key encryption scheme are CCA-secure, then the resulting hybrid scheme is CCA-secure as well.

Interestingly, Kurosawa and Desmedt have recently shown [25] that the public-key encapsulation scheme does not necessarily need to be CCA-secure in order for the resulting hybrid scheme to be CCA-secure. In particular, they show a hybrid encryption scheme which is based on, but more efficient than, the Cramer-Shoup scheme [13] *when used for hybrid encryption.* The specific hybrid schemes proposed here are as efficient as the Kurosawa-Desmedt scheme in terms of encryption (and, in one case, key generation), but somewhat less efficient in other measures; we provide detailed comparisons in Section 4. It is somewhat surprising that constructions based on completely different approaches end up having such similar performance for both encryption and key generation.

1.3 Outline

In Section 3, we present and prove secure a generic construction of a CCA-secure encryption scheme based on a variety of primitives (IBE, MACs, and encapsulation) formally defined in Section 2. Section 4 describes in more detail two specific instantiations of the various primitives; the efficiency of the resulting schemes are then compared with previous work.

2 Basic Definitions

We review the standard definitions of public-key encryption schemes and their security against adaptive chosen-ciphertext attacks. This is followed by definitions of identity-based encryption, message authentication, and "encapsulation" as needed for our construction.

Definition 1 (Public-key encryption) A public-key encryption scheme PKE is a triple of PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 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 Enc takes as input a public key PK and a message $m \in \{0,1\}^*$, and outputs a ciphertext C. We write $C \leftarrow \mathsf{Enc}_{PK}(m)$.
- The decryption algorithm Dec 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 $m \leftarrow \mathsf{Dec}_{SK}(C)$.

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

Definition 2 (CCA security) A public-key encryption scheme PKE is secure against adaptive chosen-ciphertext attacks (i.e., is "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 $\mathsf{Dec}_{SK}(\cdot)$.
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" $C^* \leftarrow \mathsf{Enc}_{PK}(m_b)$.
4. A may continue to query its decryption oracle $\mathsf{Dec}_{SK}(\cdot)$ except that it may not request the decryption of C^*.
5. Finally, A outputs a guess b'.

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

2.1 Identity-Based Encryption

Informally, an IBE scheme is a public-key encryption scheme in which any string (i.e., identity) can serve as a public key. In more detail, a setup algorithm is first run to generate "master" public and secret keys. Given the master secret key and any string $ID \in \{0,1\}^*$ (which can be viewed as an identity), it is possible to derive a "personal secret key" SK_{ID}. Any sender can encrypt a message for "identity" ID 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 multiple identities $ID' \neq ID$. The concept of identity-based encryption was introduced by Shamir [31], and provably-secure IBE schemes in the random oracle model were demonstrated by Boneh and Franklin [7] and Cocks [12]. More recently, provably-secure IBE schemes in the standard model have been developed [10, 5, 6, 34]; see further discussion below.

In the original definition of security for IBE proposed and achieved by Boneh and Franklin [7], the adversary may choose the "target identity" (ID in the above discussion) in an adaptive manner, based on the master public key and any keys $SK_{ID'}$ the adversary has obtained thus far. A weaker notion of security, proposed and achieved by Canetti, Halevi, and Katz [10], 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. As in [11], our construction only requires weak IBE schemes secure against chosen-plaintext attacks. We therefore only recall this definition of security.

Definition 3 (IBE) An identity-based encryption scheme IBE is a 4-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ 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 Enc 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 $C \leftarrow \mathsf{Enc}_{PK}(ID, m)$.

- The decryption algorithm Dec 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 $m \leftarrow \mathsf{Dec}_{SK_{ID}}(ID, C)$.

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 $\mathsf{Enc}_{PK}(ID, m)$ we have $\mathsf{Dec}_{SK_{ID}}(ID, C) = m$. ■

As mentioned earlier, we provide a definition of security only for the case of "weak" IBE, as considered in [10, 5]. (Of course, a scheme satisfying the stronger definition of [7, 6] is trivially a weak IBE scheme as well.)

Definition 4 (Selective-ID IBE) 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" $C^* \leftarrow \mathsf{Enc}_{PK}(ID^*, m_b)$.

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 $|\Pr_{A,\mathsf{IBE}}[\mathsf{Succ}] - 1/2|$. ■

For completeness, we remark that a slightly weaker definition — in which $\ell = \Omega(\log k)$ is *a priori* bounded, rather than being given as a parameter to Setup — suffices for our construction.

2.2 Message Authentication

We view a *message authentication code* as a pair of PPT algorithms $(\mathsf{Mac}, \mathsf{Vrfy})$. The authentication algorithm Mac takes as input a key sk and a message M, and outputs a string tag. The verification algorithm Vrfy takes as input a key sk, a message M, and a string tag; it outputs either 0 ("reject") or 1 ("accept"). We require that for all sk and M we have $\mathsf{Vrfy}_{sk}(M, \mathsf{Mac}_{sk}(M)) = 1$. For simplicity, we assume that Mac and Vrfy are deterministic.

We give a definition of security tailored to the requirements of our construction; in particular, we require only "one-time" security for our message authentication code. We remark that efficient schemes satisfying this definition can be constructed without any computational assumptions using, e.g., almost strongly universal hash families [35].

Definition 5 (Message authentication) A message authentication code $(\mathsf{Mac}, \mathsf{Vrfy})$ is secure against a one-time chosen-message attack if the success probability of any PPT adversary A in the following game is negligible in the security parameter k:

1. A random key $sk \in \{0,1\}^k$ is chosen.
2. $A(1^k)$ outputs a message M and is given in return $\mathsf{tag} = \mathsf{Mac}_{sk}(M)$.
3. A outputs a pair (M', tag').

We say that A *succeeds* if $(M, \mathsf{tag}) \neq (M', \mathsf{tag}')$ and $\mathsf{Vrfy}_{sk}(M', \mathsf{tag}') = 1$. ■

In the above, the adversary succeeds even if $M = M'$ but $\mathsf{tag} \neq \mathsf{tag}'$. Thus, the definition corresponds to what has been termed "strong" security in the context of signature schemes.

2.3 Encapsulation

We define a notion of "encapsulation" which may be viewed as a weak variant of commitment. (Note that our definition is unrelated to that of *key encapsulation* which was discussed in Section 1.2.) In terms of functionality, an encapsulation scheme commits the sender to a *random string* as opposed to a chosen message as in the case of commitment. In terms of security, our construction only requires binding to hold for *honestly-generated encapsulations*; this is analogous to assuming an honest sender during the first phase of a commitment scheme.

Definition 6 (Encapsulation) An encapsulation scheme is a triple of PPT algorithms $(\mathsf{Setup}, \mathcal{S}, \mathcal{R})$ such that:

- Setup takes as input the security parameter 1^k and outputs a string pub.
- $\mathcal{S}$ takes as input 1^k and pub, and outputs $(r, \mathsf{com}, \mathsf{dec})$ with $r \in \{0,1\}^k$. We refer to com as the public commitment string and dec as the de-commitment string.
- $\mathcal{R}$ takes as input $(\mathsf{pub}, \mathsf{com}, \mathsf{dec})$ and outputs an $r \in \{0,1\}^k \cup \{\perp\}$.

We require that for all pub output by Setup and for all $(r, \mathsf{com}, \mathsf{dec})$ output by $\mathcal{S}(1^k, \mathsf{pub})$, we have $\mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{dec}) = r$. We also assume for simplicity that com and dec have fixed lengths for any given value of the security parameter. ■

As in the case of commitment, an encapsulation scheme satisfies notions of both binding and hiding. Informally, "hiding" requires that com should leak no information about r; more formally, the string r should be indistinguishable from random even when given com (and pub). "Binding" requires that an honestly-generated com can be "opened" to only a single (legal) value of r; see below.

Definition 7 (Secure encapsulation) An encapsulation scheme $(\mathsf{Setup}, \mathcal{S}, \mathcal{R})$ is secure if it satisfies both hiding and binding as follows:

Hiding: The following is negligible for all PPT A:

$$\left|\Pr\left[\begin{array}{c}\mathsf{pub} \leftarrow \mathsf{Setup}(1^k); r_0 \leftarrow \{0,1\}^k;\\ (r_1, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(1^k, \mathsf{pub}); b \leftarrow \{0,1\}\end{array} : A(1^k, \mathsf{pub}, \mathsf{com}, r_b) = b\right] - \frac{1}{2}\right|.$$

Binding: The following is negligible for all PPT A:

$$\Pr\left[\begin{array}{c}\mathsf{pub} \leftarrow \mathsf{Setup}(1^k);\\ (r, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(1^k, \mathsf{pub});\\ \mathsf{dec}' \leftarrow A(1^k, \mathsf{pub}, r, \mathsf{com}, \mathsf{dec})\end{array} : \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{dec}') \notin \{\perp, r\}\right].$$

■

In the above, both hiding and binding are required to hold only computationally. In Section 4 we show a novel encapsulation scheme which is both simple and efficient, and which achieves *statistical* hiding (and computational binding).

3 A Generic Construction

We now describe our construction of a CCA-secure encryption scheme from the primitives introduced in the previous section. Let $(\mathsf{Setup}', \mathsf{Der}', \mathsf{Enc}', \mathsf{Dec}')$ be an IBE scheme, $(\mathsf{Setup}, \mathcal{S}, \mathcal{R})$ be an encapsulation scheme, and $(\mathsf{Mac}, \mathsf{Vrfy})$ be a message authentication code. Our scheme is constructed as follows:

Key generation Keys for our scheme are generated by running $\mathsf{Setup}'(1^k)$ to generate (PK, msk) and $\mathsf{Setup}(1^k)$ to generate pub. The public key is (PK, pub), and the secret key is msk.

Encryption To encrypt a message m using public key (PK, pub), a sender first encapsulates a random value by running $\mathcal{S}(1^k, \mathsf{pub})$ to obtain $(r, \mathsf{com}, \mathsf{dec})$. The sender then encrypts the "message" $m \circ \mathsf{dec}$ with respect to the "identity" com; that is, the sender computes $C \leftarrow \mathsf{Enc}'_{PK}(\mathsf{com}, m \circ \mathsf{dec})$. The resulting ciphertext C is then authenticated by using r as a key for a message authentication code; i.e., the sender computes $\mathsf{tag} = \mathsf{Mac}_r(C)$. The final ciphertext is $\langle \mathsf{com}, C, \mathsf{tag} \rangle$.

Decryption To decrypt a ciphertext $\langle \mathsf{com}, C, \mathsf{tag} \rangle$, the receiver derives the secret key SK_{com} corresponding to the "identity" com, and uses this key to decrypt the ciphertext C as per the underlying IBE scheme; this yields a "message" $m \circ \mathsf{dec}$ (if decryption fails, the receiver outputs $\perp$). Next, the receiver runs $\mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{dec})$ to obtain a string r; if $r \neq \perp$ and $\mathsf{Vrfy}_r(C, \mathsf{tag}) = 1$, the receiver outputs m. Otherwise, the receiver outputs $\perp$.

Intuition for the security of the above encryption scheme against chosen-ciphertext attacks is similar to [11]. Let $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$ be the challenge ciphertext (cf. Definition 2). In the absence of any decryption queries, it is clear that the value of the bit b remains hidden from the adversary due to the security of the underlying IBE scheme. Decryption queries of the form $\langle \mathsf{com}, C, \mathsf{tag} \rangle$ with $\mathsf{com} \neq \mathsf{com}^*$ do not further help the adversary since the adversary would be unable to determine b

even if it had the secret key SK_{com} *corresponding to* com (this follows again from the security of the underlying IBE scheme). Thus, it is left to examine decryption queries of the form $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$. The crux of our proof is to show that all queries of this form are rejected (i.e., the decryption oracle returns $\perp$ in response to all queries of this form) with all but negligible probability. A formal proof of this statement is somewhat involved, as it requires avoiding the apparent "circularity" arising from the IBE scheme, the message authentication code, and the encapsulation scheme; the details are given in the proof below.

Theorem 1 *Assuming the IBE scheme, message authentication code, and encapsulation scheme used above satisfy Definitions 2.1, 2.2, and 2.3, respectively, the above construction is a* PKE *scheme which is secure against adaptive chosen-ciphertext attacks.*

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

Let $\ell(k)$ denote the length of strings com output by $\mathcal{S}$. Define adversary $\mathcal{A}'$ as follows:

1. $\mathcal{A}'(1^k, \ell(k))$ runs $\mathsf{Setup}(1^k)$ to generate pub, and runs $\mathcal{S}(1^k, \mathsf{pub})$ to obtain $(r^*, \mathsf{com}^*, \mathsf{dec}^*)$. The adversary $\mathcal{A}'$ then outputs the "target identity" com^*.

2. A' is then given IBE parameters PK. Adversary A', in turn, runs $\mathcal{A}$ on inputs 1^k and (PK, pub).

3. When $\mathcal{A}$ submits the ciphertext $\langle \mathsf{com}, C, \mathsf{tag}\rangle$ to its decryption oracle, $\mathcal{A}'$ proceeds as follows:

   - If $\mathsf{com} = \mathsf{com}^*$, then $\mathcal{A}'$ returns $\perp$.
   - If $\mathsf{com} \neq \mathsf{com}^*$, then $\mathcal{A}'$ makes the oracle query $\mathsf{Der}'_{\mathsf{msk}}(\mathsf{com})$ to obtain SK_{com}. It then computes $m \circ \mathsf{dec} = \mathsf{Dec}'_{SK_{\mathsf{com}}}(\mathsf{com}, C)$, followed by $r = \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{dec})$. If $r \neq \perp$ and $\mathsf{Vrfy}_r(C, \mathsf{tag}) = 1$, it returns m to A. Otherwise, it returns $\perp$.

4. At some point, $\mathcal{A}$ outputs two messages m_0, m_1. Adversary A' outputs the messages $m_0 \circ \mathsf{dec}^*$ and $m_1 \circ \mathsf{dec}^*$, and receives in return a ciphertext C^*. It computes $\mathsf{tag}^* = \mathsf{Mac}_{r^*}(C^*)$ and returns $\langle \mathsf{com}^*, C^*, \mathsf{tag}^*\rangle$ to 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 strategy for attacking the underlying IBE scheme in a selective-identity, chosen-plaintext attack; in particular, $\mathcal{A}'$ never requests the secret key corresponding to "target identity" com^*.

Before analyzing the success probability of A', we prove a claim bounding the probability of a certain event. Say a ciphertext $\langle \mathsf{com}, C, \mathsf{tag}\rangle$ is *valid* if decryption of this ciphertext would not result in $\perp$. Let Valid denote the event that A ever submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ to its decryption oracle which is valid. (We always implicitly assume that $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle \neq \langle \mathsf{com}^*, C^*, \mathsf{tag}^*\rangle$ since this event is disallowed after A is given the challenge ciphertext, and occurs with only negligible probability before A is given the challenge ciphertext.)

Claim $\Pr[\mathsf{Valid}]$ *is negligible.*

Proof Let Game 0 denote the original experiment in which A interacts with a real decryption oracle (and not the simulated decryption oracle provided by A'); we are interested in bounding $\Pr_0[\mathsf{Valid}]$. Let Equiv be the event that the adversary ever submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ for which (1) C decrypts to some arbitrary $m \circ \mathsf{dec}$ (using the secret key SK_{com^*}) and furthermore (2) $\mathcal{R}(\mathsf{pub}, \mathsf{com}^*, \mathsf{dec}) = r$ with $r \notin \{r^*, \perp\}$. Let Forge be the event that Equiv does *not* occur, and A at some point submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ such that $\mathsf{Vrfy}_{r^*}(C, \mathsf{tag}) = 1$. Clearly, we have $\Pr_0[\mathsf{Valid}] \leq \Pr_0[\mathsf{Equiv}] + \Pr_0[\mathsf{Forge}]$.

We first show that $\Pr_0[\mathsf{Equiv}]$ is negligible, by the binding property of the encapsulation scheme. Consider an adversary B acting as follows: given input $(1^k, \mathsf{pub}, r^*, \mathsf{com}^*, \mathsf{dec}^*)$, adversary B generates (PK, msk) for the IBE scheme and runs A on inputs 1^k and (PK, pub). Whenever A makes a decryption oracle query, B can legitimately answer this query since B knows msk. When A submits its two messages m_0, m_1, adversary B simply chooses $b \in \{0,1\}$ at random and encrypts m_b in the expected way to generate a completely valid ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$ (B can easily do this since it has both r^* and dec^*). Now, if Equiv ever occurs then B learns dec such that $\mathcal{R}(\mathsf{pub}, \mathsf{com}^*, \mathsf{dec}) \notin \{\perp, r^*\}$. But this exactly violates the binding property of $(\mathsf{Setup}, \mathcal{S}, \mathcal{R})$.

We next show that $\Pr_0[\mathsf{Forge}]$ is negligible. Let $q(k)$ be a polynomial upper bound on the number of decryption queries made by A, and let Forge_i denote the event that Forge occurs for the first time on the i^{th} decryption query of A. Let Forge'_i denote the event that the i^{th} decryption query is of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ and $\mathsf{Vrfy}_{r^*}(C, \mathsf{tag}) = 1$ *when all previous decryption queries of the form* $\langle \mathsf{com}^*, C', \mathsf{tag}' \rangle$ *are answered with* $\perp$ (without checking whether they are valid or not). We refer to this latter "game" (which formally depends on the i under consideration) as Game $0'$.

Note that $\Pr_0[\mathsf{Forge}] = \sum_{i=1}^{q(k)} \Pr_0[\mathsf{Forge}_i]$. Furthermore, for all i we have $\Pr_{0'}[\mathsf{Forge}'_i] \geq \Pr_0[\mathsf{Forge}_i]$. Letting $\mathsf{Forge}' \stackrel{\text{def}}{=} \cup_i \mathsf{Forge}'_i$, we obtain $\Pr_0[\mathsf{Forge}] \leq \Pr_{0'}[\mathsf{Forge}']$.

Define Game 1 which proceeds exactly as Game $0'$, except that A is now given a random encryption of $m_b \circ 0^{n(k)}$ instead of a random encryption of $m_b \circ \mathsf{dec}^*$ (here, $n(k) \stackrel{\text{def}}{=} |\mathsf{dec}^*|$; recall that Definition 2.3 requires the length of dec^* to be fixed for a given value of k). We claim that $\left|\Pr_{0'}[\mathsf{Forge}'] - \Pr_1[\mathsf{Forge}']\right|$ is negligible. Indeed, if this is not the case then we can easily construct an algorithm B attacking the security of the underlying IBE scheme:

- Given input 1^k, algorithm B runs $\mathsf{Setup}(1^k)$ to generate pub and then runs $\mathcal{S}(1^k, \mathsf{pub})$ to obtain $(r^*, \mathsf{com}^*, \mathsf{dec}^*)$. It outputs com^* as the target identity and is then given the IBE parameters PK. Finally, it runs A on inputs 1^k and (PK, pub).

- Decryption queries of A are answered as follows:

  - Queries of the form $\langle \mathsf{com}, C, \mathsf{tag} \rangle$ with $\mathsf{com} \neq \mathsf{com}^*$ are answered by first querying $\mathsf{Der}'_{\mathsf{msk}}(\mathsf{com})$ to obtain SK_{com}, and then decrypting in the usual way.
  - Upon receiving a query of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$, first check whether $\mathsf{Vrfy}_{r^*}(C, \mathsf{tag}) = 1$. If so, abort the experiment and output 1. Otherwise, return $\perp$ to A.

- Eventually, A sends a pair of messages m_0, m_1 to its encryption oracle. B selects a bit b at random, and sends $m_b \circ \mathsf{dec}^*$ and $m_b \circ 0^{n(k)}$ to its encryption oracle. It receives in return a challenge ciphertext C^*, and uses this to generate a ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$ in the natural way.

- Further decryption queries of A are answered as above.

- If A halts and B has not previously aborted the experiment, then B outputs a random bit.

The probability that B outputs 1 when given an encryption of $m_b \circ \mathsf{dec}^*$ is $\frac{1}{2} + \frac{1}{2} \cdot \Pr_{0'}[\mathsf{Forge}']$. On the other hand, the probability that B outputs 1 when given an encryption of $m_b \circ 0^{n(k)}$ is $\frac{1}{2} + \frac{1}{2} \cdot \Pr_1[\mathsf{Forge}']$. Since the difference between these two probabilities must be negligible if the underlying IBE scheme is secure, this proves the current claim.

Define Game 2 which proceeds exactly as Game 1, except that the challenge ciphertext given to A is now constructed as follows: $\mathcal{S}(1^k, \mathsf{pub})$ is run to give $(r, \mathsf{com}^*, \mathsf{dec}^*)$ but an independent random key $r^* \in \{0,1\}^k$ is chosen as well. Compute $C^* \leftarrow \mathsf{Enc}_{PK}(\mathsf{com}^*, m \circ 0^{n(k)})$, followed by $\mathsf{tag}^* = \mathsf{Mac}_{r^*}(C^*)$. The challenge ciphertext, as usual, is $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$. We claim that the difference $\left|\Pr_1[\mathsf{Forge}'] - \Pr_2[\mathsf{Forge}']\right|$ is negligible. To see this, consider the following algorithm B breaking the hiding property of the encapsulation scheme:

- B is given input 1^k and $(\mathsf{pub}, \mathsf{com}^*, \tilde{r})$. It then runs $\mathsf{Setup}'(1^k)$ to generate (PK, msk), and runs A on input 1^k and (PK, pub).
- Decryption queries of A are answered as follows:
  - Queries of the form $\langle \mathsf{com}, C, \mathsf{tag} \rangle$ with $\mathsf{com} \neq \mathsf{com}^*$ are answered by running $\mathsf{Der}'_{\mathsf{msk}}(\mathsf{com})$ to obtain SK_{com}, and then decrypting in the usual way.
  - Upon receiving a query of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$, first check whether $\mathsf{Vrfy}_{\tilde{r}}(C, \mathsf{tag}) = 1$. If so, abort the experiment and output 1. Otherwise, return $\perp$ to A.
- Eventually, A sends a pair of messages m_0, m_1 to its encryption oracle. B selects a bit b at random and proceeds as follows: it computes $C^* \leftarrow \mathsf{Enc}_{PK}(\mathsf{com}^*, m_b \circ 0^{n(k)})$, computes $\mathsf{tag}^* = \mathsf{Mac}_{\tilde{r}}(C^*)$, and returns the challenge ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$ to A.
- Further decryption queries of A are answered as above.
- If A halts and B has not previously aborted the experiment, then B outputs a random bit.

Now, if $\tilde{r}$ is such that $(\tilde{r}, \mathsf{com}^*, \mathsf{dec}^*)$ was output by $\mathcal{S}(1^k, \mathsf{pub})$ then the view of A is exactly as in Game 1 and so the probability that B outputs 1 in this case is $\frac{1}{2}(1 + \Pr_1[\mathsf{Forge}'])$. On the other hand, if $\tilde{r}$ is chosen independently of com^* then the view of A is exactly as in Game 2 and so the probability that B outputs 1 in this case is $\frac{1}{2}(1 + \Pr_2[\mathsf{Forge}'])$. Since the difference between these two probabilities must be negligible by the hiding property of the encapsulation scheme, this proves the current claim.

Finally, we claim that $\Pr_2[\mathsf{Forge}']$ is negligible. This follows quite easily from the security of the message authentication code, and we omit the details here. This completes the proof of the claim. ■

Given the preceding claim, we see that the simulation which A' provides for A is statistically close to a real execution of A: in particular, the only difference occurs when Valid occurs. We therefore conclude that the advantage of A' is negligibly close to the advantage of A. Since the advantage of A' is negligible under the assumed security of the underlying IBE, the advantage of A must be negligible as well. This completes the proof of Theorem 1. ■

4 Efficient Instantiations

Here, we describe two particular instantiations of our scheme by describing specific instantiations of the various primitives.

IBE schemes. Boneh and Boyen [5] recently proposed two efficient IBE schemes suitable for our purposes. We refer to [5] for the full details and content ourselves with giving only a high-level description of their first scheme here. Let $\mathbb{G}$ and $\mathbb{G}_1$ be two (multiplicative) cyclic groups of prime order q for which there exists an efficiently-computable map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ which is *bilinear* and *non-degenerate*. Namely, (1) for all $\mu, \nu \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$ we have $\hat{e}(\mu^a, \nu^b) = e(\mu, \nu)^{ab}$, and (2) $\hat{e}(g, g) \neq 1$ for some generator g of $\mathbb{G}$. The IBE scheme is defined as follows:

Setup Pick random generators g, g_1, g_2 of $\mathbb{G}$ and a random $x \in \mathbb{Z}_q$. Set $g_3 = g^x$ and $Z = \hat{e}(g_1, g_3)$. The master public key is $PK = (g, g_1, g_2, g_3, Z)$ and the master secret key is $\mathsf{msk} = x$.

Derive To derive the secret key for "identity" $ID \in \mathbb{Z}_q$ using $\mathsf{msk} = x$, choose a random $r \in \mathbb{Z}_q$ and return the key $SK_{ID} = (g_1^x g_2^r g_3^{r \cdot ID},\ g^r)$.

Encrypt To encrypt a message $M \in \mathbb{G}_1$ with respect to "identity" $ID \in \mathbb{Z}_q$, choose a random $s \in \mathbb{Z}_q$ and output the ciphertext $(g^s,\ g_2^s g_3^{s \cdot ID},\ M \cdot Z^s)$.

Decrypt To decrypt ciphertext (A, B, C) using private key (K_1, K_2), output $C \cdot \hat{e}(B, K_2)/\hat{e}(A, K_1)$.

Correctness can be easily verified. Security of the above scheme is based on the decisional bilinear Diffie-Hellman (decision-BDH) problem. For efficiency, we assume that the master secret key msk contains the discrete logarithms of g_1, g_2, and g_3 with respect to base g, in which case generating SK_{ID} requires only two exponentiations.

The second IBE scheme of Boneh and Boyen [5] is more efficient than the above in terms of both key-generation and decryption time (the time required for encryption is essentially the same), but is based on a cryptographic assumption which is less standard.

When the above scheme is used for key encapsulation (in the sense of Section 1.2), the sender need only send $(g^s,\ g_2^s g_3^{s \cdot ID})$ and compute the key $H_\alpha(Z^s)$ where H is a keyed hash function (see below); the receiver, given ciphertext $\langle A, B \rangle$, computes the matching key $H_\alpha(\hat{e}(A, K_1)/\hat{e}(B, K_2))$, where K_1, K_2 are as before. In this description, H represents a keyed hash function where the key α is included as part of the receiver's public key. Under the decisional-BDH assumption, it suffices for H to be chosen from a pairwise-independent hash family in order for the scheme to be secure. We remark, however, that this encapsulation scheme is also secure under a potentially weaker "hash BDH" assumption as well (and a similar remark holds also for the second IBE scheme of [5]). See further discussion at the end of this section.

Message authentication codes. A number of efficient message authentication codes are known, and we do not suggest any particular one. We stress that we only require "one-time" security (cf. Definition 2.2) and so efficient schemes which do not rely on any computational assumptions (e.g., [35]) may be used. Furthermore, messages to be authenticated have a (known) fixed length; this enables slight optimizations and/or simplifications of known schemes.

Encapsulation schemes. We suggest an encapsulation scheme based on a fixed cryptographic hash function $H : \{0,1\}^{448} \to \{0,1\}^{128}$ (constructed, e.g., by suitably modifying the output length of SHA-1), and for a particular choice of security parameters; it is easy to adapt the scheme for the more general case. Our scheme works as follows:

- Setup chooses a hash function h from a family of pairwise independent hash functions mapping 448-bit strings to 128-bit strings, and outputs $\mathsf{pub} = h$.

- The encapsulation algorithm $\mathcal{S}$ takes pub as input, chooses a random $x \in \{0,1\}^{448}$, and then outputs $(r = h(x), \mathsf{com} = H(x), \mathsf{dec} = x)$.

- The recovery algorithm $\mathcal{R}$ takes as input $(\mathsf{pub} = h, \mathsf{com}, \mathsf{dec})$ and outputs $h(\mathsf{dec})$ if $H(\mathsf{dec}) = \mathsf{com}$, and $\perp$ otherwise.

Note that binding holds as long as it is infeasible to find a $\mathsf{dec}' \neq \mathsf{dec}$ such that $H(\mathsf{dec}') = H(\mathsf{dec})$, *where* dec *is chosen uniformly at random* (cf. Definition 2.3). Thus, binding holds as long as H is second-preimage resistant (the construction can be easily modified so as to be based on UOWHFs by simply having Setup choose a key h' for a UOWHF and including h' in pub); collision-resistance is not necessary.[3] Furthermore, the above scheme satisfies statistical hiding. More specifically:

Theorem 2 *For the encapsulation scheme described above, the statistical difference between the following distributions is at most* 2^{-63}*:*

$$\begin{array}{ll} (1) & \{\mathsf{pub} \leftarrow \mathsf{Setup}; (r, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(\mathsf{pub}) : (\mathsf{pub}, \mathsf{com}, r)\} \\ (2) & \{\mathsf{pub} \leftarrow \mathsf{Setup}; (r, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(\mathsf{pub}); r' \leftarrow \{0,1\}^{128} : (\mathsf{pub}, \mathsf{com}, r')\}. \end{array}$$

Proof (Sketch) The idea is loosely based on [16, 21], but our proof is much simpler. For any $x \in \{0,1\}^{448}$, let $N_x \stackrel{\text{def}}{=} \{x' \mid H(x') = H(x)\}$ (this is simply the set of elements hashing to $H(x)$). Call x *good* if $|N_x| \geq 2^{255}$, and *bad* otherwise. Since the output length of H is 128 bits, there are at most $2^{255} \cdot 2^{128} = 2^{383}$ bad x's; thus, the probability that an x chosen uniformly at random from $\{0,1\}^{448}$ is bad is at most 2^{-65}.

Assuming x is good, the min-entropy of x — given pub and com — is at least 255 bits since every $\tilde{x} \in N_x$ is equally likely. Viewing h as a strong extractor (or, equivalently, applying the leftover-hash lemma [22]) we see that $\{h, H(x), h(x)\}$ has statistical difference at most 2^{-64} from $\{h, H(x), U_{128}\}$, where U_{128} represents the uniform distribution over $\{0,1\}^{128}$. The theorem follows easily. ■

A concrete scheme. Given the primitives above, we may construct a CCA-secure encryption scheme as described in the previous section. However, as discussed in Section 1.2, improved efficiency can be obtained by directly constructing a hybrid encryption scheme; we do so here.

Key generation requires running the key-generation algorithm for the underlying IBE scheme and then choosing a hash function h from a family of pairwise independent hash functions.

Encryption of a message M involves (1) running the encapsulation scheme to obtain $(k = h(x), ID = H(x), x)$; (2) using the underlying IBE as a key encapsulation scheme, with identity ID, to generate a ciphertext C_1 encapsulating a key k'; (3) using k' to encrypt $M \circ x$ by, for example, computing $C_2 = G(k') \oplus (M \circ x)$, where G is a PRG; (4) computing a MAC on C_1, C_2 using key k.

The ciphertext consists of ID, C_1, C_2, and the tag output by the MAC.

Decryption of ciphertext $(ID, C_1, C_2, \mathsf{tag})$ is done in the obvious way: recover k' from C_1 (using identity ID), recover $M \circ x$ from C_2, and compute $k = h(x)$. If $H(x) = ID$ and $\mathsf{Vrfy}_k((C_1, C_2), \mathsf{tag}) = 1$, then output M; otherwise, output $\perp$.

We tabulate the efficiency of our schemes, and compare them to the scheme of Kurosawa-Desmedt [25], in Table 1. Scheme 1 is instantiated using the first IBE from [5], as described above;

[3] This also explains why an output length of 128 bits for H should provide a sufficient level of security.

	Encryption	Decryption	Key generation	Ciphertext overhead
Scheme 1	3.5 p-exps.	2 p-exps. + 2 pairings	3 exps.	$2\lvert p\rvert + 704$
Scheme 2	3.5 p-exps.	1.5 exps. + 1 pairing	2 exps.	$2\lvert p\rvert + 704$
KD [25]	3.5 p-exps.	1.5 exps.	3 exps.	$2\lvert p\rvert + 128$

Table 1: Efficiency comparison for CCA-secure hybrid encryption schemes. When tabulating computational efficiency, "private-key" operations (hash function/block cipher evaluations) are ignored, and one multi-exponentiation is counted as 1.5 exponentiations. Ciphertext overhead represents the difference (in bits) between the ciphertext length and the message length, and $|p|$ is the length (in bits) of a group element. "p-exp" refers to an exponentiation relative to a fixed base.

scheme 2 is instantiated using the second IBE from [5]. During encryption all bases of exponentiation are fixed which potentially enables further speed-up by pre-computation. In Scheme 1 we assume that g_1, g_3 are generated by raising the fixed generator g to a random power. Hence, computing $\hat{e}(g_1, g_3)$ requires only a single exponentiation assuming $\hat{e}(g, g)$ is pre-computed.

In addition to comparing the efficiency of these various schemes, it is interesting also to compare the cryptographic assumptions on which they are based. Security of the Kurosawa-Desmedt scheme (as in the case of the Cramer-Shoup scheme [13] on which it is based) *inherently* relies on the decisional Diffie-Hellman assumption, and it does not seem possible to obtain provable security using a weaker variant of this assumption. In contrast, as noted earlier, our schemes may be proven secure under "hash BDH"-type assumptions which are potentially weaker than the decisional-BDH assumption.[4]

5 Conclusions

We present an efficient methodology for constructing CCA-secure public-key cryptosystems from weak identity-based encryption schemes. Our construction adds only a MAC and a weak "commitment" to the original IBE system. Consequently, performance of the resulting public-key system is very close to the performance of the underlying IBE scheme. This improves on a previous transformation of Canetti, et al. which relies on the use of one-time signature schemes.

Applying our construction to recent IBE systems of Boneh and Boyen we obtain an efficient CCA-secure public-key cryptosystem without random oracles. Encryption (and, in one case, key generation) in the resulting systems are more efficient than in the Cramer-Shoup scheme, and on par with the recent proposal of Kurosawa and Desmedt. Decryption time and ciphertext size are comparable, though a bit worse. Our schemes are also somewhat more flexible than the Kurosawa-Desmedt scheme in terms of the cryptographic assumptions needed to obtain a proof of security. Our results show that building CCA-secure systems from IBE can produce very efficient schemes. The resulting schemes, as well as the proofs of security, are very different from those based on the work of Cramer and Shoup.

[4]In fact, we may base security of our constructions on purely *computational* — rather than *decisional* — assumptions; e.g., the computational-BDH assumption (using hard-core bits to encrypt one bit at a time). Although this no longer yields a practical scheme, it achieves CCA-secure encryption based on a computational assumption while avoiding the extreme inefficiency of NIZK proofs.

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 Comm. Security*, ACM, pp. 62–73, 1993.

[3] D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS#1. *Adv. in Cryptology — Crypto 1998*, LNCS vol. 1462, Springer-Verlag, pp. 1–12, 1998.

[4] 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.

[5] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. *Adv. in Cryptology — Eurocrypt 2004*, LNCS vol. 3027, Springer-Verlag, pp. 223–238, 2004. Full version available from `http://eprint.iacr.org/2004/172`

[6] D. Boneh and X. Boyen. Secure Identity Based Encryption Without Random Oracles. *Adv. in Cryptology — Crypto 2004*, LNCS vol. 3152, Springer-Verlag, pp. 443–459, 2004.

[7] 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 in *SIAM J. Computing* 32(3): 586–615, 2003 and available from `http://crypto.stanford.edu/~dabo/pubs.html`

[8] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. *42nd IEEE Symp. on Foundations of Computer Science (FOCS)*, IEEE, pp. 136–145, 2001. Full version available at `http://eprint.iacr.org/2000/067/`

[9] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. *30th ACM Symp. on Theory of Computing (STOC)*, ACM, pp. 209–218, 1998.

[10] 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`

[11] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *Adv. in Cryptology — Eurocrypt 2004*, LNCS vol. 3027, Springer-Verlag, pp. 207–222, 2004.

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

[13] 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.

[14] 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.

[15] 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.

[16] I. Damgård, T.P. Pedersen, and B. Pfitzmann. On the Existence of Statistically-Hiding Bit Commitment Schemes and Fail-Stop Signatures. *Adv. in Cryptology — Crypto 1993*, LNCS vol. 773, Springer-Verlag, pp. 250–265, 1993.

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

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

[19] 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.

[20] S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. *SIAM J. Computing* 17(2): 281–308, 1988.

[21] S. Halevi and S. Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. *Adv. in Cryptology — Crypto 1996*, LNCS vol. 1109, Springer-Verlag, pp. 201–215, 1996.

[22] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. Construction of a Pseudorandom Generator from any One-Way Function. *SIAM J. Comp.* 28(4): 1364–1396, 1999.

[23] N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, and W. Whyte. The Impact of Decryption Failures on the Security of NTRU Encryption. *Adv. in Cryptology — Crypto 2003*, LNCS vol. 2729, Springer-Verlag, pp. 226–246, 2003.

[24] M. Joye, J.-J. Quisquater, and M. Yung. On the Power of Misbehaving Adversaries and Security Analysis of the Original EPOC. *Cryptographers' Track — RSA 2001*, LNCS vol. 2020, Springer-Verlag, pp. 208–222, 2001.

[25] K. Kurosawa and Y. Desmedt. A New Paradigm of Hybrid Encryption Scheme. *Adv. in Cryptology — Crypto 2004*, LNCS vol. 3152, Springer-Verlag, pp. 426–442, 2004.

[26] L. Lamport. Constructing Digital Signatures from a One-Way Function. Technical Report CSL-98, SRI International, 1978.

[27] 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.

[28] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. *21st ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 33–43, 1989.

[29] 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.

[30] 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.

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

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

[33] V. Shoup. Using Hash Functions as a Hedge Against Chosen Ciphertext Attack. *Adv. in Cryptology — Eurocrypt 2000*, LNCS vol. 275–288, Springer-Verlag, pp. 1807, 2000.

[34] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. Available at `http://eprint.iacr.org/2004/180`

[35] M.N. Wegman and J.L. Carter. New Hash Functions and Their Use in Authentication and Set Equality. *J. Computer System Sciences* 22(3): 265–279, 1981.