Chosen-Ciphertext Security from Identity-Based Encryption*

Dan Boneh† Ran Canetti‡ Shai Halevi§ Jonathan Katz¶

June 13, 2006

Abstract

We propose simple and efficient CCA-secure public-key encryption schemes (i.e., schemes secure against adaptive chosen-ciphertext attacks) based on any identity-based encryption (IBE) scheme. Our constructions have ramifications of both theoretical and practical interest. First, our schemes give a new paradigm for achieving CCA-security; this paradigm avoids "proofs of well-formedness" that have been shown to underlie previous constructions. Second, instantiating our construction using known IBE constructions we obtain CCA-secure encryption schemes whose performance is competitive with the most efficient CCA-secure schemes to date.

Our techniques extend naturally to give an efficient method for securing IBE schemes (even hierarchical ones) against adaptive chosen-ciphertext attacks. Coupled with previous work, this gives the first efficient constructions of CCA-secure IBE schemes.

Keywords: identity-based encryption, public-key encryption, chosen-ciphertext security

1 Introduction

Security against adaptive chosen-ciphertext attacks [49, 50, 28, 2] is the *de facto* level of security required for public-key encryption schemes used in practice. This security notion is appropriate for encryption schemes used in the presence of *active* attackers who may potentially modify messages in transit, and schemes proven secure with respect to this notion may be securely "plugged in" to higher-level protocols deployed in unauthenticated networks that were designed and analyzed under the idealized assumption of "secure channels" (see, e.g., [15, 19]). Unfortunately, only a handful of approaches are known for constructing encryption schemes that meet this notion of security. In this work we put forward a new approach for constructing such schemes.

---

*This work appeared in preliminary form as *Chosen-ciphertext security from identity-based encryption*, in Eurocrypt 2004 [18] and *Improved efficiency for CCA-secure cryptosystems built using identity-based encryption*, in RSA-CT 2005 [12].

†Department of Computer Science, Stanford University, Stanford, CA 94305 (dabo@cs.stanford.edu). This research was supported by NSF grant CNS-0331640.

‡IBM T. J. Watson Research Center, Hawthorne, NY 10532 (canetti@watson.ibm.com). This research was supported by NSF grant CNS-0430450.

§IBM T. J. Watson Research Center, Hawthorne, NY 10532 (shaih@alum.mit.edu).

¶Department of Computer Science, University of Maryland, College Park, MD 20742 (jkatz@cs.umd.edu). This research was supported by NSF grant CNS-0310751.

1.1 Background

Security for public-key encryption was first defined formally by Goldwasser and Micali [38]. Their notion of *semantic security*, roughly speaking, requires that observation of a ciphertext does not enable an adversary to compute anything about the underlying plaintext message that it could not have computed on its own (i.e., prior to observing the ciphertext); this should hold even if the adversary has some *a priori* information about the message. Goldwasser and Micali (see also [47, 35, 36]) proved that semantic security is equivalent to the notion of *indistinguishability* that requires (roughly) the following: for any two messages, given a "challenge" ciphertext C that is an encryption of one of these messages it is infeasible to determine (with any noticeable advantage over a random guess) which message was actually encrypted. Because these definitions imply security even when the adversary can mount a *chosen-plaintext attack* to obtain encryptions of messages of its choice, we will refer to these notions using the commonly-accepted term "CPA-security."

CPA-security does not guarantee any security against *chosen-ciphertext attacks* by which an adversary may obtain decryptions of ciphertexts of its choice (note that attacks of this sort may arise in practice [5, 53]). Indistinguishability-based definitions appropriate for this setting were given by Naor and Yung [49] and Rackoff and Simon [50]. Naor and Yung consider a *non-adaptive* chosen-ciphertext attack in which the adversary may request decryptions only *before* it obtains the challenge ciphertext. Rackoff and Simon define the stronger notion of security against *adaptive* chosen-ciphertext attacks whereby the adversary may request decryptions even after seeing the challenge ciphertext, under the natural limitation that the adversary may not request decryption of the challenge ciphertext itself. (We will refer to the latter notion as "CCA-security.") See [28, 2, 36] for further discussion of these definitions. Extensions of semantic security to the case of chosen-ciphertext attacks were considered in [36, 37, 55], where it is shown that, as in the case of CPA-security, these definitions are equivalent to the indistinguishability-based ones.

As we have already mentioned, CCA-security is now the *de facto* level of security for public-key encryption due to its numerous advantages (some of which were summarized earlier). Unfortunately, only a handful of public-key encryption schemes have been proven secure against adaptive chosen-ciphertext attacks without resorting to heuristics such as the random oracle methodology [3], a controversial and problematic approach [16].

In fact, prior to this work only two approaches were known for constructing CCA-secure cryptosystems. The first follows the paradigm introduced by Naor and Yung [49] to achieve non-adaptive chosen-ciphertext security, later extended to the case of adaptive chosen-ciphertext security by Dolev, Dwork, and Naor [28] and Sahai [51]. This technique uses as building blocks any CPA-secure public-key encryption scheme and any non-interactive zero-knowledge (NIZK) proof system for all of $\mathcal{NP}$ [6, 31]. Consequently, this approach can be based on general cryptographic assumptions [31]: specifically, the existence of enhanced trapdoor permutations [36, Sect. C.4.1]. Encryption schemes resulting from this approach, however, are highly impractical because they employ generic NIZK proofs which in turn require a Karp reduction from the $\mathcal{NP}$ language of interest to some $\mathcal{NP}$-complete language. Thus, given current state of the art, this approach serves as a feasibility result for the existence of CCA-secure cryptosystems based on general assumptions but does not lead to any practical constructions.

The second technique is due to Cramer and Shoup [21, 22], and is based on algebraic constructs with particular homomorphic properties (namely, those admitting "smooth hash proof systems" in the terminology of [22]). Algebraic constructs of the appropriate type are known to exist based on some specific number-theoretic assumptions [21, 22], including the decisional Diffie-Hellman (DDH) assumption. Other constructions relying on this technique have been given recently [32, 23, 42], leading to a number of practical schemes.

Interestingly, Elkind and Sahai have observed [29] that both the above approaches for constructing CCA-secure encryption schemes can be viewed as special cases of a single paradigm. In this paradigm one starts with a CPA-secure cryptosystem in which certain "ill-formed" ciphertexts are indistinguishable from honestly-generated ciphertexts. A CCA-secure cryptosystem is then obtained by having the sender honestly generate a ciphertext using the underlying CPA-secure scheme, and then append a "proof of well-formedness" (satisfying certain criteria) to this ciphertext. The NIZK proofs used by Sahai [51] as well as the smooth hash proof systems used by Cramer and Shoup [21, 22] are shown by Elkind and Sahai to satisfy the appropriate criteria.

1.2 Summary of Our Results

We propose two new approaches for constructing CCA-secure public-key encryption schemes based on any CPA-secure identity-based encryption (IBE) scheme[1] (see Sections 2.1 and 3.1 for definitions of the latter notion). A number of IBE schemes based on specific number-theoretic assumptions are known [17, 7, 8, 56, 34]; thus, our techniques yield new constructions of CCA-secure encryption schemes based on these same assumptions.

From a theoretical perspective, our work offers new ways of constructing CCA-secure encryption schemes that do not use "proofs of well-formedness" and hence do not seem to fit within the Elkind-Sahai characterization mentioned above. From a practical perspective, we show that a specific instantiation of our construction yields a practical CCA-secure scheme with efficiency close to that of the best previous construction (cf. Section 7). This efficient instantiation is based on the *decisional bilinear Diffie-Hellman (BDH) assumption*, described in Section 7.3. Comparing this assumption to those used in prior constructions of CCA-secure encryption schemes, we note that:

- The decisional BDH assumption seems incomparable to the assumption of enhanced trapdoor permutations that underlies the standard construction of generic NIZK as used in the schemes of [28, 51]. Nevertheless, the decisional BDH assumption is known to imply the existence of generic NIZK proof systems for all of $\mathcal{NP}$ [17, Appendix B], and hence was already known to imply CCA-secure encryption via [28, 51]. We stress again that the schemes shown here are orders of magnitude more efficient than schemes constructed via generic NIZK.

- The decisional BDH assumption implies the decisional Diffie-Hellman assumption that underlies the schemes of [21, 42]. However, less efficient variants of our scheme can be proven secure using assumptions that are incomparable to the decisional Diffie-Hellman assumption; see footnote 6 in Section 7.3. In addition, our scheme lends itself to threshold encryption much more efficiently than previous schemes; see below.

Further extensions and applications. Both our approaches extend to give a transformation from any CPA-secure $(\ell + 1)$-level hierarchical identity-based encryption (HIBE) scheme [41, 33] to a CCA-secure ℓ-level HIBE scheme (HIBE is described in Section 3.2). In particular, applying our technique to any 2-level HIBE scheme gives a CCA-secure IBE scheme. Using this approach with known HIBE schemes [17, 7, 8, 56, 9] yields the first efficient constructions of CCA-secure IBE schemes.

[1] Our constructions use other primitives, but these can all be constructed based on one-way functions which are in turn implied by CPA-secure encryption.

Our first approach, when instantiated with an appropriate IBE scheme, serves as the basis for the first CCA-secure threshold encryption scheme with *non-interactive* decryption [7, 10]. (In a threshold encryption scheme [25] the secret key is shared among multiple servers, some fraction of whom must cooperate in order to decrypt a given ciphertext.) Security in this case crucially relies on specific properties of our construction, and in particular the feature that a certain class of "valid" ciphertexts can be efficiently recognized without knowledge of the global secret key; the reader is referred to [7, 10] for further discussion.

Our approaches are generic and can be used to construct a CCA-secure encryption scheme from an *arbitrary* IBE scheme. Extending our work, Boyen et al. [14] show that for some concrete IBE schemes (e.g., the one of Waters [56]) a more efficient and direct construction of a CCA-secure encryption scheme is possible.

1.3 Organization

In the following section, we provide an informal overview of identity-based encryption and hierarchical identity-based encryption, as well as some high-level intuition regarding our techniques. Formal definitions of all relevant cryptographic notions (including IBE and CCA-secure public-key encryption) appear in Section 3 and Appendix A. The first of our transformations is discussed in Section 4, and the second transformation is presented in Section 5. We discuss the extension to HIBE in Section 6.

The treatment in the above sections is generic, and does not rely on any specific cryptographic assumptions. In Section 7 we recall one specific number-theoretic assumption under which an IBE scheme is known to exist, describe a concrete instantiation of our construction based on this assumption, and compare the efficiency of the resulting CCA-secure encryption scheme to the most efficient such construction that was previously known (namely, the Kurosawa-Desmedt variant [42] of the Cramer-Shoup encryption scheme [21]).

2 Overview of Our Techniques

2.1 Identity-Based Encryption

Before sketching our constructions, we first recall the notion of IBE as introduced by Shamir [52]. 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 trusted authority called a *private-key generator* (PKG) is assumed to initialize the system by running a key-generation algorithm to generate "master" public and secret keys. The master public key PK is published, while the PKG stores the master secret key. Given the master secret key and an arbitrary string ID (viewed as the identity of a party in the system), the PKG can derive a "personal secret key" SK_{ID} and give it to this party. Any sender can encrypt a message for this party using only the master public key PK and the string ID; we denote such encryption by $\mathcal{E}_{PK}(ID, \cdot)$. The resulting ciphertext can be decrypted using the personal secret key SK_{ID}, but the following extension of CPA-security is required to hold:

> For any two messages and any identity ID, given a challenge ciphertext C that is an encryption of one of these messages (with respect to ID) it is infeasible to determine (with any noticeable advantage over a random guess) which message was actually encrypted. *This should hold even if the adversary is given $SK_{ID'}$ for multiple identities $ID' \neq ID$ chosen adaptively by the adversary.*

The first formal definition of security for IBE was given by Boneh and Franklin [11]. In their definition, the adversary may choose the "target identity" (ID in the above) in an adaptive manner, based on the master public key PK and any keys $\{SK_{ID'}\}$ the adversary has obtained thus far; we call such schemes "fully secure." A weaker notion, proposed by Canetti, et al. [17] and called "selective-ID" security there, requires the adversary to specify the target identity *in advance*, before the master public key is published. Fully-secure IBE schemes in the random oracle model were first demonstrated by Boneh and Franklin [11] and Cocks [20]. Canetti, et al. [17], building on earlier work of Gentry and Silverberg [33], constructed an IBE scheme satisfying selective-ID security in the standard model; more efficient constructions were given by Boneh and Boyen [7]. More recently, Boneh and Boyen [8] have shown a fully-secure IBE scheme in the standard model, and more efficient constructions were subsequently given by Waters [56] and Gentry [34].

Both our constructions of CCA-secure encryption from IBE require an IBE scheme satisfying only the weaker notion of selective-ID security. Our transformation from any CPA-secure $(\ell+1)$-level HIBE scheme to a CCA-secure ℓ-level HIBE scheme preserves the level of security in the above sense: i.e., if the original scheme is fully secure then so is the derived scheme, but selective-ID security of the original scheme is sufficient for selective-ID security of the derived scheme.

2.2 Our Techniques

Our first construction. Given an IBE scheme, we construct a CCA-secure public-key encryption scheme as follows: The public key of the new scheme is the master public key PK of the IBE scheme and the secret key is the corresponding master secret key. To encrypt a message with respect to public key PK, the sender first generates a key-pair (vk, sk) for a strong[2] one-time signature scheme, and then encrypts the message with respect to the "identity" vk. The resulting ciphertext $C \leftarrow \mathcal{E}_{PK}(vk, m)$ 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 using 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. We may first notice that any valid ciphertext $c = \langle vk, C, \sigma \rangle$ submitted by the adversary to its decryption oracle (implying $c \neq c^*$) must, except with negligible probability, have $vk \neq vk^*$ by the strong security of the one-time signature scheme. The crux of the security proof is then to show that (selective-ID) security of the IBE scheme implies that obtaining the decryption of C does not help the adversary in deciding which message the ciphertext C^* corresponds to. Intuitively, this is because the adversary cannot guess the message corresponding to C^* with probability better than $1/2$ *even if it were given the secret key* SK_{vk}. (This is so since $vk \neq vk^*$, and C^* was encrypted for "identity" vk^* using an IBE scheme.) But giving SK_{vk} to the adversary only makes the adversary more powerful, since it could then decrypt C itself.

Our use of a strong one-time signature scheme to force the adversary's decryption queries to differ from the challenge ciphertext in a specific way is reminiscent of prior work in the context

[2]A "strong" signature scheme has the property that it is infeasible to create a new, valid signature even for a previously-signed message. A formal definition is given in Appendix A.

of CCA-security [28, 51]. The key difference is that prior work used the verification key vk to implement "unduplicatable set selection" (cf. [51]) which requires $\Theta(k)$ invocations of some underlying encryption scheme, where k is the security parameter. Furthermore, prior work also required some sort of "proof of consistency" for the resulting ciphertext, leading (as described earlier) to an impractical scheme. In contrast, our construction gives a CCA-secure encryption scheme with relatively minimal overhead as compared to the original IBE scheme.

We note also independent work of MacKenzie, et al. [45], who introduce a weaker notion of CCA-secure encryption and use essentially the same construction to convert any scheme satisfying their weaker definition into a full-fledged CCA-secure encryption scheme. Their work, however, only shows efficient realizations of schemes in the random oracle model.

We remark that if the signature scheme used is only unforgeable in the standard sense (rather than *strongly* unforgeable) we obtain an encryption scheme satisfying the slightly weaker notion of *replayable* CCA-security [19]. Also, a simple modification of the above construction gives an encryption scheme secure against *non-adaptive* chosen-ciphertext attacks [49, 28, 2] but with essentially no overhead as compared to the underlying IBE scheme. Namely, replace the verification key vk by a randomly-chosen string $r \in \{0,1\}^{\omega(\log k)}$; 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 (with better than negligible probability) which r will be used for the challenge ciphertext, an argument similar to the above shows that this scheme is secure against non-adaptive chosen-ciphertext attacks.

Improving the efficiency. Focusing again on security against *adaptive* chosen-ciphertext attacks, the previous construction — although conceptually simple and efficient — does add noticeable overhead in practice to the underlying IBE scheme: encryption requires the sender to generate signing/verification keys and sign a message; the ciphertext length is increased by the size of a verification key plus the size of a signature; and decryption requires the receiver to perform a signature verification. Although one-time signatures are "easy" to construct in theory, and are more efficient than full-fledged signatures (i.e., those which are strongly unforgeable under adaptive chosen-message attack), they still have their price. In particular:

- Known one-time signature schemes based on general one-way functions [43, 30] allow very efficient *signing*; key generation and signature verification, on the other hand, require $\Theta(k)$ evaluations of the one-way function and are relatively expensive. More problematic, perhaps, is that such schemes have very long public keys and/or signatures (with combined length $\Theta(k^2)$), resulting in very long ciphertexts in our construction above.

- One-time signature schemes can of course be based on number-theoretic assumptions (say, by adapting full-fledged signature schemes); this yields schemes whose computational cost for key generation, signing, and verifying is more expensive, but which (may) have the advantage of short(er) public keys and signatures.

Motivated by the above, we modify the previous construction by using a message authentication code (MAC) in place of a one-time signature scheme in the following way: the secret signing key is replaced by a secret MAC key r and the public verification key (which was used as the "identity" for the IBE scheme) is replaced by a commitment to r. In more detail, encryption of a message m is performed by first committing to a random MAC key r, resulting in a commitment com and a corresponding decommitment dec. The ciphertext is $\langle \mathsf{com}, C, \mathsf{tag}\rangle$, where C is an encryption of the "message" $m \circ \mathsf{dec}$ with respect to the "identity" com (i.e., $C \leftarrow \mathcal{E}_{PK}(\mathsf{com}, m \circ \mathsf{dec})$) and tag is a message authentication code computed on C using key r. Decryption of ciphertext $\langle \mathsf{com}, C, \mathsf{tag} \rangle$ is done in the natural way: the receiver first decrypts C with respect to "identity" com to obtain $m \circ \mathsf{dec}$, and then recovers r using com and dec. The receiver then tries to verify tag using key r, outputting m if verification succeeds and $\perp$ otherwise.

In fact, a weaker form of commitment than the standard one suffices for our purposes (cf. Section 5.1); we refer to this weaker notion as "encapsulation." The advantage of the former is that encapsulation schemes can potentially be more efficient than full-fledged commitment schemes.

The intuition for the security of this construction is quite similar to that discussed previously. Consider a challenge ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$ that was constructed using MAC key r^*. As before, decryption queries that use a different "identity" $\mathsf{com} \neq \mathsf{com}^*$ are useless to the attacker due to the security of the underlying identity-based encryption scheme. For decryption queries $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ that use the same "identity" we note that either (1) C decrypts to $m \circ \mathsf{dec}$, where dec is a valid decommitment of com^* to some $r \neq r^*$, or (2) the attacker was able to compute a valid tag on $C \neq C^*$ with respect to the MAC key r^*. The first case is easily shown to violate the binding property of the commitment scheme. The second case can be shown to violate either the secrecy of the commitment scheme (i.e., the adversary learns something about r^* from com^*), the secrecy of the encryption (i.e., the adversary learns something about r^* from C^*), or the security of the MAC (i.e., the adversary generates a valid tag without learning anything about r^*).

The actual proof for this scheme is more difficult than for the previous case due to the fact that here C must be decrypted *before* validity of the ciphertext as a whole can be checked. We thus must be careful to avoid the seeming circularity which arises since the MAC key r is used to authenticate a string (namely, C) that depends on r (via dec).

The idea of using a message authentication code and a commitment to the key was suggested previously in the context of non-malleable commitment (e.g., [26, 27]), but our application of this technique is *qualitatively* different precisely due to the apparent circularity (and the resulting complications to the proof) discussed above. In particular, in the context of non-malleable commitment the MAC key can be revealed by the sender during the decommitment phase and hence the key is not used to authenticate a message which depends on itself. In contrast, here the MAC key must be transmitted to the receiver as part of the ciphertext. The idea of encapsulating the MAC key (rather than using full-fledged commitment), as well as the encapsulation scheme we propose, are new to this work.

3 Definitions

We use the standard definitions of public-key encryption schemes and their security against adaptive chosen-ciphertext attacks, and strong one-time signature schemes and message authentication codes. For convenience and to fix notation, we recall these definitions in Appendix A. Our definitions of IBE and HIBE schemes have also appeared previously; however, since these definitions are less familiar yet are central to our work, we include the appropriate definitions in this section. Encapsulation schemes are defined in Section 5.1. Our definitions and proofs are phrased with respect to uniform adversaries but can be easily extended to the non-uniform setting. We let "PPT" stand for "probabilistic polynomial-time."

If Σ is a set then Σ^n denotes the set of n-tuples of elements of Σ, with Σ^0 denoting the set containing only the empty tuple. Thus, using this notation, $\{0,1\}^n$ denotes the set of binary strings of length n. We also define $\Sigma^{<n} \stackrel{\text{def}}{=} \bigcup_{0 \leq i < n} \Sigma^i$ and $\Sigma^{\leq n} \stackrel{\text{def}}{=} \bigcup_{0 \leq i \leq n} \Sigma^i$.

3.1 Identity-Based Encryption

We begin by reviewing the functional definition of an IBE scheme [11].

Definition 1 An *identity-based encryption scheme* for identities of length n (where n is a polynomially-bounded function) is a 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. It outputs a master public key PK and a master secret key msk. (We assume that k and $n = n(k)$ are implicit in PK and msk.)
- The (possibly randomized) *key-derivation algorithm* Der takes as input the master secret key msk and an identity $ID \in \{0,1\}^n$. 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 master public key PK, an identity $ID \in \{0,1\}^n$, and a message m in some implicit[3] message space; it outputs a ciphertext C. We write $C \leftarrow \mathcal{E}_{PK}(ID, m)$.
- The (possibly randomized) *decryption algorithm* $\mathcal{D}$ takes as input an identity ID, an associated decryption key SK_{ID}, and a ciphertext C. It outputs a message m or the symbol $\perp$ (which is not in the message space). We write $m \leftarrow \mathcal{D}_{SK_{ID}}(ID, C)$.

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

We now define security for identity-based encryption. As mentioned in the Introduction, the definition we give is weaker than that considered by Boneh and Franklin [11] and conforms to "selective-ID" security [17] where the "target" identity is selected by the adversary before the public key is generated.

Definition 2 An identity-based encryption scheme Π for identities of length n is *selective-ID secure against chosen-plaintext attacks* if the advantage of any PPT adversary $\mathcal{A}$ in the following game is negligible in the security parameter k:

1. $\mathcal{A}(1^k)$ outputs a "target" identity $ID^* \in \{0,1\}^{n(k)}$.
2. $\mathsf{Setup}(1^k)$ outputs (PK, msk). The adversary is given PK.
3. The adversary $\mathcal{A}$ may make polynomially-many queries to an oracle $\mathsf{Der}_{\mathsf{msk}}(\cdot)$, except that it may not request a secret key corresponding to the target identity ID^*.

   The adversary is allowed to query this oracle repeatedly *using the same identity*; if Der is randomized, then a different secret key may possibly be returned each time.
4. At some point, $\mathcal{A}$ 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. $\mathcal{A}$ may continue to query its oracle $\mathsf{Der}_{\mathsf{msk}}(\cdot)$ as above. Finally, $\mathcal{A}$ outputs a guess b'.

[3]For example, the message space may consist of all strings of length $p(k)$, where p is polynomially-bounded.

We say that $\mathcal{A}$ *succeeds* if $b' = b$, and denote the probability of this event by $\Pr^{\mathsf{IBE}}_{\mathcal{A},\Pi}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Pi}(k) \stackrel{\text{def}}{=} \left|\Pr^{\mathsf{IBE}}_{\mathcal{A},\Pi}[\mathsf{Succ}] - 1/2\right|$. $\diamondsuit$

The definition may be extended to take into account security against adaptive chosen-ciphertext attacks. In this case, the adversary 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 $\mathsf{Der}_{\mathsf{msk}}(ID^*)$).[4] The adversary has access to this oracle throughout the entire game, but cannot submit the challenge ciphertext C^* to $\widehat{\mathcal{D}}$.

On deterministic vs. randomized key derivation: For simplicity, when dealing with chosen-ciphertext security for IBE schemes we will assume that Der is *deterministic*. If Der is not deterministic, a definition of chosen-ciphertext security is complicated by the question of whether different invocations of the decryption oracle $\widehat{\mathcal{D}}$ should use the *same* secret key SK_{ID^*} (computed using Der the first time $\widehat{\mathcal{D}}$ is invoked) or a *fresh* secret key (computed by running Der using fresh random coins each time). The resulting security definitions obtained in each case seem incomparable, and there does not appear to be any reason to prefer one over the other. A related difficulty arises in the case of hierarchical identity-based encryption (discussed next) even in the case of chosen-plaintext attacks. These distinctions all become irrelevant when Der is deterministic.

Assuming deterministic key derivation is anyway without much loss of generality: given an IBE scheme with randomized key-derivation algorithm Der we can construct an IBE scheme with deterministic key derivation by (1) including a random key sk for a pseudorandom function F as part of the master secret key msk; and (2) generating the decryption key for identity ID by running $\mathsf{Der}_{\mathsf{msk}}(ID)$ using "randomness" $F_{sk}(ID)$. A similar idea applies to the case of HIBE.

3.2 Hierarchical Identity-Based Encryption

Hierarchical identity-based encryption (HIBE) is an extension of IBE suggested by Horwitz and Lynn [41]. In an ℓ-level HIBE scheme, there is again assumed to be a trusted authority who generates master public and secret keys. As in the case of IBE, it is possible to derive a personal secret key SK_{ID_1} for any identity ID_1 using the master secret key. The additional functionality provided by an HIBE scheme is that this personal secret key SK_{ID_1} may now be used to derive a personal secret key SK_{ID_1,ID_2} for the "ID-vector" (ID_1, ID_2), and so on, with the scheme supporting the derivation of keys in this way for ID-vectors of length at most ℓ. As in the case of IBE, any sender can encrypt a message for the ID-vector $v = (ID_1, \ldots, ID_L)$ using only the master public key and v; the resulting ciphertext can be decrypted by anyone who knows $SK_{ID_1,\ldots,ID_L}$. Security is defined as the natural analogue of security in the case of IBE: informally, indistinguishability should hold for ciphertexts encrypted with respect to a target ID-vector $v = (ID_1, \ldots, ID_L)$ as long as the adversary does not know the secret keys of any identity of the form $(ID_1, \ldots, ID_{L'})$ for $L' \leq L$.

Before formally defining an ℓ-level HIBE scheme, we first introduce some notation to deal with ID-vectors $v \in (\{0,1\}^n)^{\leq \ell}$. For an ID-vector $v = (v_1, \ldots, v_L)$ (with $v_i \in \{0,1\}^n$), we define the *length of* v as $|v| = L$ and let $v.r$ (for $r \in \{0,1\}^n$) denote the ID-vector $(v_1, \ldots, v_L, r)$ of length $|v|+1$. We let ε denote the ID-vector of length 0. Given v as above and an ID-vector $v' = (v'_1, \ldots, v'_{L'})$, we say that v *is a prefix of* v' if $|v| \leq |v'|$ and $v_i = v'_i$ for $i \leq |v|$.

Re-phrased using the above notation, the functional property of an ℓ-level HIBE scheme is this: given the secret key SK_v associated with the ID-vector v it is possible to derive a secret key

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

$SK_{v'}$ associated with the ID-vector v' (assuming $|v'| \leq \ell$) whenever v is a prefix of v'. Similarly, the security provided by an HIBE scheme is that indistinguishability should hold for ciphertexts encrypted with respect to an ID-vector v even if the adversary has multiple keys $\{SK_{v'}\}_{v' \in V}$ for some set V *as long as no* $v' \in V$ *is a prefix of* v.

Formal definitions follow. The functional definition is essentially from [33], although we assume for simplicity that the key derivation algorithm is deterministic (cf. the remark in the previous section). As in the case of IBE, the definition of security we give is the one proposed by Canetti, et al. [17], which is weaker than the one considered in [33].

Definition 3 An *ℓ-level HIBE scheme for identities of length n* (where ℓ, n are polynomially-bounded functions) is a 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. It outputs a master public key PK and a master secret key denoted SK_ε. (We assume that $k, \ell = \ell(k)$, and $n = n(k)$ are implicit in PK and all node secret keys.)
- The deterministic *key-derivation algorithm* Der takes as input an ID-vector $v \in (\{0,1\}^n)^{<\ell}$, its associated secret key SK_v, and a string $r \in \{0,1\}^n$. It returns the secret key $SK_{v.r}$ associated with the ID-vector $v.r$. We write this as $SK_{v.r} := \mathsf{Der}_{SK_v}(v, r)$.
- The randomized *encryption algorithm* $\mathcal{E}$ takes as input the master public key PK, an ID-vector $v \in (\{0,1\}^n)^{\leq \ell}$, and a message m in some implicit message space. It outputs a ciphertext C. We write this as $C \leftarrow \mathcal{E}_{PK}(v, m)$.
- The (possibly randomized) *decryption algorithm* $\mathcal{D}$ takes as input an ID-vector $v \in (\{0,1\}^n)^{\leq \ell}$, its associated secret key SK_v, and a ciphertext C. It returns a message m or the symbol $\perp$ (which is not in the message space). We write $m \leftarrow \mathcal{D}_{SK_v}(v, C)$.

We require that for all (PK, SK_ε) output by Setup, all $v \in (\{0,1\}^n)^{\leq \ell}$, any secret key SK_v correctly generated (in the obvious way) for v, and any message m we have $m = \mathcal{D}_{SK_v}(v, \mathcal{E}_{PK}(v, M))$. $\diamondsuit$

Definition 4 An ℓ-level HIBE scheme Π for identities of length n is *selective-ID secure against chosen-plaintext attacks* if the advantage of any PPT adversary $\mathcal{A}$ in the following game is negligible in the security parameter k:

1. Let $\ell = \ell(k)$, $n = n(k)$. Adversary $\mathcal{A}(1^k)$ outputs a "target" ID-vector $v^* \in (\{0,1\}^n)^{\leq \ell}$.
2. Algorithm $\mathsf{Setup}(1^k)$ outputs (PK, SK_ε). The adversary is given PK.
3. The adversary may adaptively ask for the secret key(s) corresponding to any ID-vector(s) v, as long as v is not a prefix of the target ID-vector v^*. The adversary is given the secret key SK_v correctly generated for v using SK_ε and (repeated applications of) Der.
4. At some point, the adversary 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 $C^* \leftarrow \mathcal{E}_{PK}(v^*, m_b)$.
5. The adversary can continue asking for secret keys as above. Finally, $\mathcal{A}$ outputs a guess b'.

We say that $\mathcal{A}$ *succeeds* if $b' = b$, and denote the probability of this event by $\Pr^{\mathsf{HIBE}}_{\mathcal{A},\Pi}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $\mathsf{Adv}^{\mathsf{HIBE}}_{\mathcal{A},\Pi}(k) \stackrel{\text{def}}{=} \left|\Pr^{\mathsf{HIBE}}_{\mathcal{A},\Pi}[\mathsf{Succ}] - 1/2\right|$. $\diamondsuit$

As in the case of IBE, it is easy to modify the above to take into account security against adaptive chosen-ciphertext attacks. Here, the adversary may additionally query an oracle $\widehat{\mathcal{D}}(\cdot,\cdot)$ such that $\widehat{\mathcal{D}}(v, C)$ returns $\mathcal{D}_{SK_v}(v, C)$ using key SK_v correctly generated for v. The only restriction is that the adversary may not query $\widehat{\mathcal{D}}(v^*, C^*)$ after receiving the challenge ciphertext C^*.

4 Chosen-Ciphertext Security from Identity-Based Encryption

Given an IBE scheme $\Pi' = (\mathsf{Setup}, \mathsf{Der}, \mathcal{E}', \mathcal{D}')$ for identities of length n which is selective-ID secure against chosen-plaintext attacks, we construct a public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathcal{E}, \mathcal{D})$ secure against adaptive chosen-ciphertext attacks. In the construction, we use a strong one-time signature scheme $\mathsf{Sig} = (\mathcal{G}, \mathsf{Sign}, \mathsf{Vrfy})$ (cf. Definition 11 in Appendix A) in which the verification key output by $\mathcal{G}(1^k)$ has length $n = n(k)$. The construction of Π proceeds as follows:

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

Encryption 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 $|vk| = n$). The sender then computes $C \leftarrow \mathcal{E}'_{PK}(vk, m)$ (i.e., the sender encrypts m with respect to the "identity" vk) and $\sigma \leftarrow \mathsf{Sign}_{sk}(C)$. The final ciphertext is $\langle vk, C, \sigma\rangle$.

Decryption 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 $m \leftarrow \mathcal{D}'_{SK_{vk}}(vk, C)$.

It is clear that the above scheme satisfies correctness. We 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 8). It should be clear that, without any decryption oracle queries, the plaintext corresponding to this ciphertext remains "hidden" to the adversary; this is so because C^* is output by Π' which is CPA-secure (and the additional components of the ciphertext provide no additional help).

We claim that decryption oracle queries cannot further help the adversary in determining the plaintext (i.e., guessing the value of b; cf. Definition 8). On one hand, if the adversary submits to its decryption oracle a ciphertext $\langle vk, C, \sigma\rangle$ that is different from the challenge ciphertext but with $vk = vk^*$ then (with all but negligible probability) 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. In the proof below, we formalize these ideas.

Theorem 1 *If Π' is an identity-based encryption scheme which is selective-ID secure against chosen-plaintext attacks and* Sig *is a strong one-time signature scheme, then Π is a public-key encryption scheme secure against adaptive chosen-ciphertext attacks.*

Proof Assume we are given a PPT adversary $\mathcal{A}$ attacking Π in an adaptive chosen-ciphertext attack. Say a ciphertext $\langle vk, C, \sigma\rangle$ is *valid* if $\mathsf{Vrfy}_{vk}(C,\sigma) = 1$. Let $\langle vk^*, C^*, \sigma^*\rangle$ denote the challenge ciphertext received by $\mathcal{A}$ during a particular run of the experiment, and let Forge denote the event that $\mathcal{A}$ submits a valid ciphertext $\langle vk^*, C, \sigma\rangle$ to the decryption oracle (we may assume that vk^* is chosen at the outset of the experiment so this event is well-defined even before $\mathcal{A}$ is given the challenge ciphertext. Recall also that $\mathcal{A}$ is disallowed from submitting the challenge ciphertext to the decryption oracle once the challenge ciphertext is given to $\mathcal{A}$.) We prove the following claims:

Claim 1 $\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}]$ *is negligible.*

Claim 2 $\left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}] - \frac{1}{2}\right|$ *is negligible.*

To see that these imply the theorem, note that

$$\begin{aligned}
&\left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ}] - \tfrac{1}{2}\right| \\
&\quad\le \left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ} \wedge \mathsf{Forge}] - \tfrac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}]\right| + \left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \tfrac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}] - \tfrac{1}{2}\right| \\
&\quad\le \tfrac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}] + \left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \tfrac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}] - \tfrac{1}{2}\right|,
\end{aligned}$$

which is negligible given the stated claims. (A concrete security bound can be derived easily.)

Proof (of Claim 1) The proof is quite straightforward. We construct a PPT forger $\mathcal{F}$ who forges a signature with respect to signature scheme Sig (in the sense of Definition 11) with probability exactly $\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}]$. Security of Sig implies the claim.

$\mathcal{F}$ is defined as follows: given input 1^k and verification key vk^* (output by $\mathcal{G}$), $\mathcal{F}$ first runs $\mathsf{Setup}(1^k)$ to obtain (PK, msk), and then runs $\mathcal{A}(1^k, PK)$. Note that $\mathcal{F}$ can answer any decryption queries of $\mathcal{A}$. If $\mathcal{A}$ happens to submit a valid ciphertext $\langle vk^*, C, \sigma\rangle$ to its decryption oracle before requesting the challenge ciphertext, then $\mathcal{F}$ simply outputs the forgery (C, σ) and stops. Otherwise, when $\mathcal{A}$ outputs messages m_0, m_1, forger $\mathcal{F}$ proceeds as follows: it chooses a random bit b, computes $C^* \leftarrow \mathcal{E}'_{PK}(vk^*, m_b)$, and obtains (from its signing oracle) a signature σ^* on the "message" C^*. Finally, $\mathcal{F}$ hands the challenge ciphertext $\langle vk^*, C^*, \sigma^*\rangle$ to $\mathcal{A}$. If $\mathcal{A}$ submits a valid ciphertext $\langle vk^*, C, \sigma\rangle$ to its decryption oracle, note that we must have $(C,\sigma) \neq (C^*, \sigma^*)$. In this case, $\mathcal{F}$ simply outputs (C,σ) as its forgery. It is easy to see that $\mathcal{F}$'s success probability (in the sense of Definition 11) is exactly $\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}]$. □

Proof (of Claim 2) We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{A}'$ which attacks the IBE scheme Π' in the sense of Definition 2. Relating the advantages of $\mathcal{A}$ and $\mathcal{A}'$ gives the desired result.

Define adversary $\mathcal{A}'$ as follows:

1. $\mathcal{A}'(1^k)$ runs $\mathcal{G}(1^k)$ to generate (vk^*, sk^*), and outputs the "target" identity $ID^* = vk^*$.
2. $\mathcal{A}'$ is given a master public key PK. Adversary $\mathcal{A}'$, in turn, runs $\mathcal{A}(1^k, PK)$.
3. When $\mathcal{A}$ makes decryption oracle query $\mathcal{D}(\langle vk, C, \sigma\rangle)$, adversary $\mathcal{A}'$ proceeds as follows:
   (a) If $vk = vk^*$ then $\mathcal{A}'$ checks whether $\mathsf{Vrfy}_{vk^*}(C,\sigma) = 1$. If so, $\mathcal{A}'$ aborts and outputs a random bit. Otherwise, it simply responds with $\perp$.
   (b) If $vk \neq vk^*$ and $\mathsf{Vrfy}_{vk}(C,\sigma) = 0$ then $\mathcal{A}'$ responds with $\perp$.

(c) If $vk \neq vk^*$ and $\mathsf{Vrfy}_{vk}(C, \sigma) = 1$, 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 responds with m.

4. At some point, $\mathcal{A}$ outputs two equal-length messages m_0, m_1. These messages are output by $\mathcal{A}'$ as well. 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 by $\mathcal{A}'$ 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'; in particular, $\mathcal{A}'$ never requests the secret key corresponding to the "target" identity vk^*. Furthermore, $\mathcal{A}'$ provides a perfect simulation for $\mathcal{A}$ until event Forge occurs. It is thus easy to see that:

$$\left|\Pr^{\mathsf{IBE}}_{\mathcal{A}',\Pi'}[\mathsf{Succ}] - \tfrac{1}{2}\right| = \left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \tfrac{1}{2}\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Forge}] - \tfrac{1}{2}\right|,$$

and the left-hand side of the above is negligible by the assumed security of Π'. □

This concludes the proof of the theorem. ■

5 A More Efficient Construction

We show here how the idea from the previous section can be implemented using a message authentication code and a primitive we call an "encapsulation scheme" instead of a one-time signature scheme. As argued in the Introduction, this results in more efficient constructions of CCA-secure encryption schemes than the previous approach. However, using MACs rather than signatures — which, in particular, will imply that ciphertext validity can no longer be determined efficiently without an appropriate decryption key — complicates the security proof somewhat.

5.1 Encapsulation

We begin by defining a notion of "encapsulation" which may be viewed as a weak variant of commitment. In terms of functionality, an encapsulation scheme commits the sender to a *random string* as opposed to a string chosen by the sender as in the case of commitment. In terms of security, encapsulation 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 5 An *encapsulation scheme* is a triple of PPT algorithms $(\mathsf{Init}, \mathcal{S}, \mathcal{R})$ such that:

- Init 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 *commitment string* and dec as the *decommitment string*.
- $\mathcal{R}$ takes as input $(\mathsf{pub}, \mathsf{com}, \mathsf{dec})$ and outputs $r \in \{0,1\}^k \cup \{\perp\}$.

We require that for all pub output by Init 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. $\diamondsuit$

As in the case of commitment, an encapsulation scheme satisfies notions of binding and hiding. Informally, "hiding" requires that com should not reveal information about r; formally, 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 6 An encapsulation scheme is *secure* if it satisfies both hiding and binding as follows:

Hiding: The following is negligible for all PPT $\mathcal{A}$:

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

Binding: The following is negligible for all PPT $\mathcal{A}$:

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

$\diamondsuit$

Both hiding and binding are required to hold only computationally. The encapsulation scheme we will later construct achieves *statistical* hiding (and computational binding).

Since encapsulation is a weaker primitive than commitment, we could use any commitment scheme as an encapsulation scheme. We will be interested, however, in optimizing the efficiency of the construction (in particular, the lengths of com and dec for a fixed value of k) and therefore focus on satisfying only the weaker requirements given above. See further discussion in Section 7.2.

5.2 The Construction

Let $\Pi' = (\mathsf{Setup}, \mathsf{Der}, \mathcal{E}', \mathcal{D}')$ be an IBE scheme for identities of length $n = n(k)$ which is selective-ID secure against chosen-plaintext attacks, let $(\mathsf{Init}, \mathcal{S}, \mathcal{R})$ be a secure encapsulation scheme in which commitments com output by $\mathcal{S}$ have length n, and let $(\mathsf{Mac}, \mathsf{Vrfy})$ be a message authentication code. We construct a public-key encryption scheme Π as follows:

Key generation Keys for our scheme are generated by running $\mathsf{Setup}(1^k)$ to generate (PK, msk) and $\mathsf{Init}(1^k)$ 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 $C \leftarrow \mathcal{E}'_{PK}(\mathsf{com}, m \circ \mathsf{dec})$. The resulting ciphertext C is then authenticated by using r as a key for a message authentication code; i.e., the sender computes $\mathsf{tag} \leftarrow \mathsf{Mac}_r(C)$. 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 $\perp$.

Theorem 2 *If Π' is an identity-based encryption scheme which is selective-ID secure against chosen-plaintext attacks, the encapsulation scheme is secure (in the sense of Definition 6), and* $(\mathsf{Mac}, \mathsf{Vrfy})$ *is a strong one-time message authentication code, then* Π *a public-key encryption scheme secure against adaptive chosen-ciphertext attacks.*

Proof Let $\mathcal{A}$ be a PPT adversary attacking Π in an adaptive chosen-ciphertext attack. On an intuitive level, the proof here is the same as the proof of Theorem 1 in the following sense: Say a ciphertext $\langle \mathsf{com}, C, \mathsf{tag}\rangle$ is *valid* if decryption of this ciphertext (using msk) does *not* result in $\perp$. Let $\langle \mathsf{com}^*, C^*, \mathsf{tag}^*\rangle$ denote the challenge ciphertext received by $\mathcal{A}$. We will show that (1) $\mathcal{A}$ submits to its decryption oracle a valid ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ (with $\langle C, \mathsf{tag}\rangle \neq \langle C^*, \mathsf{tag}^*\rangle$) only with negligible probability; and (2) assuming that the previous event does not occur, the decryption queries made by $\mathcal{A}$ do not help $\mathcal{A}$ to "learn" the underlying plaintext. The second statement is relatively easy to prove based on the security of Π'; the first, however, is now more challenging to prove since validity of a ciphertext cannot be determined without knowledge of msk. Because of this, we structure the proof as a sequence of games to make it easier to follow. We let $\Pr_i[\cdot]$ denote the probability of a particular event occurring in game i.

Game 0 is the original game in which $\mathcal{A}$ attacks Π in a chosen-ciphertext attack as described in Definition 8. Let $r^*, \mathsf{com}^*, \mathsf{dec}^*$ denote the values that are used in computing the challenge ciphertext, and notice that we may assume these values are generated at the outset of the experiment (since these values are generated independently of $\mathcal{A}$'s actions). We are interested in upper-bounding $\left|\Pr_0[\mathsf{Succ}] - \frac{1}{2}\right|$, where (recall) Succ denotes the event that $\mathcal{A}$'s output bit b' is identical to the bit b used in constructing the challenge ciphertext.

In Game 1, we modify the experiment as follows: on input a ciphertext of the form $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$, the decryption oracle simply outputs $\perp$. Let Valid denote the event that $\mathcal{A}$ submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ to its decryption oracle which is valid, and note that

$$\left|\Pr_1[\mathsf{Succ}] - \Pr_0[\mathsf{Succ}]\right| \leq \Pr_0[\mathsf{Valid}] = \Pr_1[\mathsf{Valid}].$$

The above holds since Games 0 and 1 are identical until Valid occurs.

Let NoBind denote the event that $\mathcal{A}$ at some point submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ to its decryption oracle such that: (1) C decrypts to $m \circ \mathsf{dec}$ (using the secret key SK_{com^*} derived from msk) and (2) $\mathcal{R}(\mathsf{pub}, \mathsf{com}^*, \mathsf{dec}) = r$ with $r \notin \{r^*, \perp\}$. Let Forge denote the event that $\mathcal{A}$ at some point submits a ciphertext $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ to its decryption oracle such that $\mathsf{Vrfy}_{r^*}(C, \mathsf{tag}) = 1$. We clearly have $\Pr_1[\mathsf{Valid}] \leq \Pr_1[\mathsf{NoBind}] + \Pr_1[\mathsf{Forge}]$.

It is relatively easy to see that $\Pr_1[\mathsf{NoBind}]$ is negligible assuming the binding property of the encapsulation scheme. Formally, consider an adversary $\mathcal{B}$ acting as follows: given input $(1^k, \mathsf{pub}, \mathsf{com}^*, \mathsf{dec}^*)$, adversary $\mathcal{B}$ generates (PK, msk) by running $\mathsf{Setup}(1^k)$ and then runs $\mathcal{A}$ on inputs 1^k and (PK, pub). Whenever $\mathcal{A}$ makes a query to its decryption oracle, $\mathcal{B}$ can respond to this query as required by Game 1; specifically, $\mathcal{B}$ simply responds with $\perp$ to a decryption query of the form $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$, and responds to other queries using msk. When $\mathcal{A}$ submits its two messages

m_0, m_1, adversary $\mathcal{B}$ simply chooses $b \in \{0,1\}$ at random and encrypts m_b in the expected way to generate a completely valid challenge ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$. (Note that $\mathcal{B}$ can easily do this since it has dec^* and can compute r^*.) At the end of the experiment, $\mathcal{B}$ can decrypt every query of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ that $\mathcal{A}$ made to its decryption oracle to see whether NoBind occurred and, if so, to learn a value dec such that $\mathcal{R}(\mathsf{pub}, \mathsf{com}^*, \mathsf{dec}) \notin \{r^*, \perp\}$. But this exactly violates the binding property of encapsulation scheme $(\mathsf{Init}, \mathcal{S}, \mathcal{R})$, implying that $\Pr_1[\mathsf{NoBind}]$ must be negligible.

Game 2 is derived by modifying the way the challenge ciphertext is computed. Specifically, when $\mathcal{A}$ submits its two messages m_0, m_1 we now compute $C^* \leftarrow \mathcal{E}'_{PK}(\mathsf{com}^*, 0^{|m_0|} \circ 0^n)$ followed by $\mathsf{tag}^* \leftarrow \mathsf{Mac}_{r^*}(C^*)$. The challenge ciphertext is $\langle \mathsf{com}^*, C^*, \mathsf{tag}^* \rangle$. (A random bit b is still chosen, but is only used to define event Succ.) Since the challenge ciphertext is independent of b, it follows immediately that $\Pr_2[\mathsf{Succ}] = \frac{1}{2}$.

We claim that $|\Pr_2[\mathsf{Succ}] - \Pr_1[\mathsf{Succ}]|$ is negligible. To see this, consider the following adversary $\mathcal{A}'$ attacking the IBE scheme Π' via a chosen-plaintext attack:

- Algorithm $\mathcal{A}'(1^k)$ first runs $\mathsf{Init}(1^k)$ 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 master public key PK. Finally, $\mathcal{A}'$ runs $\mathcal{A}$ on inputs 1^k and (PK, pub).
- Decryption queries of $\mathcal{A}$ are answered in the natural way:
  - Queries of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ are answered with $\perp$.
  - 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.
- Eventually, $\mathcal{A}$ submits two equal-length messages m_0, m_1. $\mathcal{A}'$ selects a bit b at random, and sends $m_b \circ \mathsf{dec}^*$ and $0^{|m_0|} \circ 0^n$ 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 $\mathcal{A}$ are answered as above.
- Finally, $\mathcal{A}$ outputs a bit b'. If $b = b'$, then $\mathcal{A}'$ outputs 0; otherwise, $\mathcal{A}'$ outputs 1.

Note that $\mathcal{A}'$ is a valid adversary. When the encryption query of $\mathcal{A}'$ is answered with an encryption of $m_b \circ \mathsf{dec}^*$, then the view of $\mathcal{A}$ is exactly as in Game 1; on the other hand, when the encryption query of $\mathcal{A}'$ is answered with an encryption of $0^{|m_0|} \circ 0^n$ then the view of $\mathcal{A}$ is exactly as in Game 2. Thus,

$$\begin{aligned} \mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A}',\Pi'}(k) &= \left| \tfrac{1}{2} \Pr_1[\mathsf{Succ}] + \tfrac{1}{2} \Pr_2[\overline{\mathsf{Succ}}] - \tfrac{1}{2} \right| \\ &= \tfrac{1}{2} \cdot \left| \Pr_1[\mathsf{Succ}] - \Pr_2[\mathsf{Succ}] \right|. \end{aligned}$$

Security of Π' implies that $\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A}',\Pi'}$ is negligible, implying that $|\Pr_2[\mathsf{Succ}] - \Pr_1[\mathsf{Succ}]|$ is negligible. An exactly analogous argument shows that $|\Pr_2[\mathsf{Forge}] - \Pr_1[\mathsf{Forge}]|$ is negligible as well. (The only difference is that $\mathcal{A}'$ runs $\mathcal{A}$ to completion and then checks whether $\mathcal{A}$ has made any decryption query of the form $\langle \mathsf{com}^*, C, \mathsf{tag} \rangle$ for which $\mathsf{Vrfy}_{r^*}(C, \mathsf{tag}) = 1$. If so, then $\mathcal{A}'$ outputs 1; otherwise, it outputs 0.)

In Game 3, we introduce one final change. The components com^* and C^* of the challenge ciphertext are computed as in Game 2; however, the component tag^* is computed by choosing a random key $r \in \{0,1\}^k$ and setting $\mathsf{tag}^* = \mathsf{Mac}_r(C^*)$. Event Forge in this game is defined as before, but using the key r; that is, Forge is now the event that $\mathcal{A}$ makes a decryption query of the form $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ for which $\mathsf{Vrfy}_r(C, \mathsf{tag}) = 1$.

We claim that $|\Pr_3[\mathsf{Forge}] - \Pr_2[\mathsf{Forge}]|$ is negligible. To see this, consider the following algorithm $\mathcal{B}$ attacking the hiding property of the encapsulation scheme:

- $\mathcal{B}$ 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 $\mathcal{A}$ on inputs 1^k and (PK, pub).
- Decryption queries of $\mathcal{A}$ are answered in the natural way.
- Eventually, $\mathcal{A}$ submits messages m_0, m_1. $\mathcal{B}$ computes $C^* \leftarrow \mathcal{E}'_{PK}(\mathsf{com}^*, 0^{|m_0|} \circ 0^n)$, computes $\mathsf{tag}^* = \mathsf{Mac}_{\tilde{r}}(C^*)$, and returns the challenge ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^*\rangle$ to $\mathcal{A}$.
- Further decryption queries of $\mathcal{A}$ are answered as above.
- When $\mathcal{A}$ halts, $\mathcal{B}$ checks whether $\mathcal{A}$ has made any decryption query of the form $\langle \mathsf{com}^*, C, \mathsf{tag}\rangle$ for which $\mathsf{Vrfy}_{\tilde{r}}(C, \mathsf{tag}) = 1$. If so, $\mathcal{B}$ outputs 1; otherwise, it outputs 0.

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 $\mathcal{A}$ is exactly as in Game 2 and so $\mathcal{B}$ outputs 1 with probability $\Pr_2[\mathsf{Forge}]$. On the other hand, if $\tilde{r}$ is chosen at random independently of com^* then the view of $\mathcal{A}$ is exactly as in Game 3 and so $\mathcal{B}$ outputs 1 with probability $\Pr_3[\mathsf{Forge}]$. The hiding property of the encapsulation scheme thus implies that $|\Pr_3[\mathsf{Forge}] - \Pr_2[\mathsf{Forge}]|$ is negligible.

To complete the proof, we show that $\Pr_3[\mathsf{Forge}]$ is negligible. This follows rather easily from the security of the message authentication code, but we sketch the details here. Let $q = q(k)$ be an upper bound on the number of decryption oracle queries made by $\mathcal{A}$, and consider the following forging algorithm $\mathcal{F}$: first, $\mathcal{F}$ chooses a random index $j \leftarrow \{1, \ldots, q\}$. Next, $\mathcal{F}$ begins simulating Game 3 for $\mathcal{A}$ in the natural way. If the j^{th} decryption query $\langle \mathsf{com}_j, C_j, \mathsf{tag}_j\rangle$ occurs before $\mathcal{A}$ makes its encryption query, then $\mathcal{F}$ simply outputs (C_j, tag_j) and halts. Otherwise, in response to the encryption query (m_0, m_1) of $\mathcal{A}$, forger $\mathcal{F}$ computes $(r^*, \mathsf{com}^*, \mathsf{dec}^*) \leftarrow \mathcal{S}(1^k, \mathsf{pub})$ followed by $C^* \leftarrow \mathcal{E}'_{PK}(\mathsf{com}^*, 0^{|m_0|} \circ 0^n)$. Next, $\mathcal{F}$ submits C^* to its Mac oracle and receives in return tag^*. Forger $\mathcal{F}$ then gives the challenge ciphertext $\langle \mathsf{com}^*, C^*, \mathsf{tag}^*\rangle$ to $\mathcal{A}$ and continues running $\mathcal{A}$ until $\mathcal{A}$ submits its j^{th} decryption query $\langle \mathsf{com}_j, C_j, \mathsf{tag}_j\rangle$. At this point, $\mathcal{F}$ outputs (C_j, tag_j) and halts.

It is not difficult to see that the success probability of $\mathcal{F}$ in outputting a valid forgery is at least $\Pr_3[\mathsf{Forge}]/q$. Since $(\mathsf{Mac}, \mathsf{Vrfy})$ is a strong one-time message authentication code and q is polynomial, this shows that $\Pr_3[\mathsf{Forge}]$ is negligible.

Putting everything together, we have:

$$\begin{aligned}
\left|\Pr_0[\mathsf{Succ}] - \tfrac{1}{2}\right| &\leq \left|\Pr_0[\mathsf{Succ}] - \Pr_1[\mathsf{Succ}]\right| + \left|\Pr_1[\mathsf{Succ}] - \tfrac{1}{2}\right| \\
&\leq \Pr_1[\mathsf{NoBind}] + \Pr_1[\mathsf{Forge}] + \left|\Pr_1[\mathsf{Succ}] - \Pr_2[\mathsf{Succ}]\right| + \left|\Pr_2[\mathsf{Succ}] - \tfrac{1}{2}\right| \\
&= \Pr_1[\mathsf{NoBind}] + \Pr_1[\mathsf{Forge}] + \left|\Pr_1[\mathsf{Succ}] - \Pr_2[\mathsf{Succ}]\right| \\
&\leq \Pr_1[\mathsf{NoBind}] + \Pr_3[\mathsf{Forge}] + \left|\Pr_2[\mathsf{Forge}] - \Pr_3[\mathsf{Forge}]\right| \\
&\quad + \left|\Pr_1[\mathsf{Forge}] - \Pr_2[\mathsf{Forge}]\right| + \left|\Pr_1[\mathsf{Succ}] - \Pr_2[\mathsf{Succ}]\right| ,
\end{aligned}$$

and all terms in the final equation are negligible. (A concrete security analysis follows easily from the above.) ■

6 Chosen-Ciphertext Security for IBE and HIBE Schemes

The techniques of the previous two sections extend relatively easily to enable construction of an ℓ-level HIBE scheme secure against chosen-ciphertext attacks based on any $(\ell+1)$-level HIBE scheme secure against chosen-plaintext attacks. (Note that an IBE scheme is simply a 1-level HIBE scheme.) We give the details for the signature-based approach of Section 4. For arbitrary $\ell \geq 1$, let $\Pi' = (\mathsf{Setup}', \mathsf{Der}', \mathcal{E}', \mathcal{D}')$ be an $(\ell+1)$-level HIBE scheme handling identities of length $n+1$, and let $\mathsf{Sig} = (\mathcal{G}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme in which the verification key output by $\mathcal{G}(1^k)$ has length $n = n(k)$. We construct an ℓ-level HIBE scheme Π handling identities of length n. The intuition behind the construction is simple: the ID-vector $v = (v_1, \ldots, v_L) \in (\{0,1\}^n)^L$ in Π will be mapped to the ID-vector

$$\mathsf{Encode}(v) \stackrel{\text{def}}{=} (0v_1, \ldots, 0v_L) \in (\{0,1\}^{n+1})^L$$

in Π'. We will maintain the invariant that the secret key SK_v for ID-vector v in Π will be the secret key $SK'_{\hat{v}}$ for ID-vector $\hat{v} = \mathsf{Encode}(v)$ in Π'. When encrypting a message m to ID-vector v in Π, the sender will generate a verification key vk and then encrypt m to the ID-vector $\hat{v}.(1vk)$ using Π'. (The resulting ciphertext will then be signed as in Section 4.) The extra 0 and 1 bits used as "padding" ensure that any decryption queries asked by an adversary (in Π) correspond (in Π') to nodes that are not ancestors of the target ID-vector.

In more detail, Π is constructed as follows:

Setup The Setup algorithm is the same as in Π'. (Note that $\mathsf{Encode}(\varepsilon) = \varepsilon$ so the master secret key $SK_\varepsilon = SK'_\varepsilon$ satisfies the desired invariant.)

Key derivation $\mathsf{Der}_{SK_v}(v, r)$ runs as follows: let $\hat{v} = \mathsf{Encode}(v)$ and $\hat{r} = \mathsf{Encode}(r)$. Run $\mathsf{Der}'_{SK_v}(\hat{v}, \hat{r})$ and output the result as $SK_{v.r}$. (To see that key derivation maintains the desired invariant given that $SK_v = SK'_{\hat{v}}$, note that $\mathsf{Encode}(v.r) = \mathsf{Encode}(v).\mathsf{Encode}(r)$.)

Encryption $\mathcal{E}_{PK}(v, m)$ first runs $\mathcal{G}(1^k)$ to obtain (vk, sk). Let $\hat{v} = \mathsf{Encode}(v).(1vk)$. The algorithm then computes $C \leftarrow \mathcal{E}'_{PK}(\hat{v}, m)$ and $\sigma \leftarrow \mathsf{Sign}_{sk}(C)$. The final ciphertext is $\langle vk, C, \sigma \rangle$.

Decryption $\mathcal{D}_{SK_v}(v, \langle vk, C, \sigma \rangle)$ proceeds as follows: first check whether $\mathsf{Vrfy}_{vk}(C, \sigma) \stackrel{?}{=} 1$. If not, output $\perp$. Otherwise, let $\hat{v} = \mathsf{Encode}(v)$ and run $\mathsf{Der}'_{SK_v}(\hat{v}, (1vk))$ to generate the key $SK^* = SK'_{\hat{v}.(1vk)}$. Then output $m := \mathcal{D}'_{SK^*}(\hat{v}, C)$.

It can be verified easily that the above scheme is correct. An analogous construction can be given using the MAC-based construction of Section 5. We now state the main result of this section:

Theorem 3 *If* Π' *is selective-ID secure against chosen-plaintext attacks and* Sig *is a strong one-time signature scheme, then* Π *is selective-ID secure against chosen-ciphertext attacks.*

Proof The proof is similar to that of Theorem 1. Given any PPT adversary $\mathcal{A}$ attacking Π in a selective-ID chosen-ciphertext attack, we define an event Forge and then prove the analogues of Claims 1 and 2 in our setting. For visual comfort, we use $\Pr[\cdot]$ instead of $\Pr^{\mathsf{HIBE}}_{\mathcal{A},\Pi}[\cdot]$.

Let v^* denote the "target" ID-vector initially output by $\mathcal{A}$, and let $\langle vk^*, C^*, \sigma^* \rangle$ be the challenge ciphertext received by $\mathcal{A}$. Let Forge be the event that $\mathcal{A}$ makes a decryption query $\widehat{\mathcal{D}}(v^*, \langle vk^*, C, \sigma \rangle)$ with $\mathsf{Vrfy}_{vk^*}(C, \sigma) = 1$. (As in the previous proof, we may assume vk^* is chosen at the beginning of the experiment and so this event is defined even before $\mathcal{A}$ receives the challenge ciphertext. Recall again that $\mathcal{A}$ is disallowed from submitting the challenge ciphertext to its decryption oracle once this ciphertext has been given to $\mathcal{A}$.) A proof exactly as in the case of Claim 1, relying again on the fact that Sig is a strong one-time signature scheme, shows that $\Pr[\mathsf{Forge}]$ is negligible.

We next show that $|\Pr[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr[\mathsf{Forge}] - \frac{1}{2}|$ is negligible. To do so, we define adversary $\mathcal{A}'$ attacking Π' in a selective-ID chosen-plaintext attack. $\mathcal{A}'$ is defined as follows:

1. $\mathcal{A}'(1^k)$ runs $\mathcal{A}(1^k)$ who, in turn, outputs an ID-vector $v^* \in (\{0,1\}^n)^{\leq \ell}$. Adversary $\mathcal{A}'$ runs $\mathcal{G}(1^k)$ to generate (vk^*, sk^*) and outputs the target ID-vector $V^* = \mathsf{Encode}(v^*).(1vk^*)$.

2. $\mathcal{A}'$ is given PK, which it gives to $\mathcal{A}$.

3. When $\mathcal{A}$ requests the secret key for ID-vector v, $\mathcal{A}'$ requests the secret key $SK'_{\hat{v}}$ for ID-vector $\hat{v} = \mathsf{Encode}(v)$ and returns this secret key to $\mathcal{A}$. Note that since v is not a prefix of the target ID-vector v^* of $\mathcal{A}$, it follows that $\hat{v}$ is not a prefix of the target ID-vector V^* of $\mathcal{A}'$.

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

   (a) If $v = v^*$ then $\mathcal{A}'$ checks whether $\mathsf{Vrfy}_{vk}(C, \sigma) = 1$. If so, then $\mathcal{A}'$ aborts and outputs a random bit. Otherwise, it simply responds with $\perp$.

   (b) If $v \neq v^*$, or if $v = v^*$ and $vk \neq vk^*$, then $\mathcal{A}'$ sets $\hat{v} = \mathsf{Encode}(v)$ and requests the secret key $SK'_{\hat{v}.(1vk)}$. (Note that $\hat{v}.(1vk)$ is not a prefix of the target ID-vector V^* of $\mathcal{A}'$, so $\mathcal{A}'$ is allowed to submit this request.) It then honestly decrypts the submitted ciphertext and returns the result to $\mathcal{A}$.

5. 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 challenge ciphertext C^*. Adversary $\mathcal{A}'$ computes $\sigma^* \leftarrow \mathsf{Sign}_{sk^*}(C^*)$ and returns challenge ciphertext $\langle vk^*, C^*, \sigma^* \rangle$ to $\mathcal{A}$.

6. Any of $\mathcal{A}$'s subsequent decryption queries, or requests for secret keys, are answered as before.

7. 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'. As in the proof of Claim 2, it follows from the security of Π' that $|\Pr[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr[\mathsf{Forge}] - \frac{1}{2}|$ is negligible. This completes the proof. ■

We remark that when Π' is fully secure against chosen-plaintext attacks [11, 33], then Π is fully secure against chosen-ciphertext attacks. A proof for this case is easily derived from the proof above.

Canetti, et al. [17] define a slightly stronger notion of HIBE which requires the HIBE scheme to support an arbitrary (polynomial) number of levels ℓ and identities of arbitrary (polynomial) length n (where ℓ, n are provided as input to the initial Setup algorithm). We refer to HIBE schemes of this type as *unbounded*. Security is defined as in Definition 4, except that the adversary's advantage must be negligible for all adversaries $\mathcal{A}$ as well as for all polynomially-bounded functions ℓ, n. Since the above construction requires only a strong one-time signature scheme, which can be constructed based on any one-way function (and hence from any secure HIBE scheme), we have the following:

Corollary 1 *If there exists an unbounded HIBE scheme which is selective-ID secure (resp., fully secure) against chosen-plaintext attacks, then there exists an unbounded HIBE scheme which is selective-ID secure (resp., fully secure) against adaptive chosen-ciphertext attacks.*

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

7 An Efficient Instantiation

Here, we describe one instantiation of our generic construction of CCA-secure cryptosystems from Section 5. We then compare the efficiency of this construction with the most efficient previously-known CCA-secure scheme. To instantiate our construction, we need to specify a message authentication code, an encapsulation scheme, and an IBE scheme which is selective-ID secure against chosen-plaintext attacks. We consider each of these in turn.

7.1 Message Authentication Code

A number of efficient (strong, one-time) message authentication codes are known. Since the computational cost of these schemes will be dominated by the computational cost of the IBE scheme, we focus instead on minimizing the lengths of the key and the tag. For concreteness, we suggest using CBC-MAC with 128-bit AES as the underlying block cipher. (In this scheme, both the secret key and the tag are 128 bits long.) We remark, however, that strong one-time MACs with information-theoretic security [57, 54] could also be used.

7.2 Encapsulation Scheme

Adapting earlier work of Damgård, et al. [24] and Halevi and Micali [39], we propose an encapsulation scheme based on any universal one-way hash function (UOWHF) family $\{H_s : \{0,1\}^{k_1} \to \{0,1\}^k\}$ (where $k_1 \geq 3k$ is a function of the security parameter k). Our scheme works as follows:

- Init chooses a hash function h from a family of pairwise-independent hash functions mapping k_1-bit strings to k-bit strings, and also chooses at random a key s defining UOWHF H_s. It outputs $\mathsf{pub} = (h, s)$.
- The encapsulation algorithm $\mathcal{S}$ takes pub as input, chooses a random $x \in \{0,1\}^{k_1}$, and then outputs $(r = h(x), \mathsf{com} = H_s(x), \mathsf{dec} = x)$.
- The recovery algorithm $\mathcal{R}$ takes as input $((h, s), \mathsf{com}, \mathsf{dec})$ and outputs $h(\mathsf{dec})$ if $H_s(\mathsf{dec}) = \mathsf{com}$, and $\perp$ otherwise.

We prove the following regarding the above scheme:

Theorem 4 *The scheme above is a secure encapsulation scheme. Specifically, the scheme is computationally binding under the assumption that $\{H_s\}$ is a UOWHF family, and statistically hiding (without any assumptions).*

Proof The binding property is easy to see. In particular, violation of the binding property implies that an adversary finds $\mathsf{dec}' \neq \mathsf{dec}$ for which $H_s(\mathsf{dec}') = H_s(\mathsf{dec})$. Since dec is chosen independently of the key s in an honest execution of $\mathcal{S}$, security of the UOWHF family implies that binding can be violated with only negligible probability. We omit the straightforward details. (Note, however, that a UOWHF rather than a collision-resistant hash function is sufficient here since the binding property we require is weaker than that required by a standard commitment scheme.)

We next prove the following claim, which immediately implies statistical hiding:

Claim 3 *For the encapsulation scheme described above, the statistical difference between the following distributions is at most* $2 \cdot 2^{\frac{2k-k_1}{3}} \leq 2 \cdot 2^{-k/3}$*:*

$$(1) \qquad \{\mathsf{pub} \leftarrow \mathsf{Setup}; (r, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(\mathsf{pub}) : (\mathsf{pub}, \mathsf{com}, r)\}$$
$$(2) \qquad \{\mathsf{pub} \leftarrow \mathsf{Setup}; (r, \mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(\mathsf{pub}); r' \leftarrow \{0,1\}^k : (\mathsf{pub}, \mathsf{com}, r')\}.$$

The proof of this claim is loosely based on [24, 39], but our proof is much simpler. Let $\alpha \stackrel{\text{def}}{=} \frac{2k_1-k}{3}$, and assume for simplicity that k_1, k are multiples of 3. Fix an arbitrary s for the remainder of the discussion. For any fixed $x \in \{0,1\}^{k_1}$, let $N_x \stackrel{\text{def}}{=} \{x' \mid H_s(x') = H_s(x)\}$; this is simply the set of elements hashing to $H_s(x)$. Call x *good* if $|N_x| \geq 2^\alpha$, and *bad* otherwise. Since the output length of H_s is k bits, there are at most $2^\alpha \cdot 2^k = 2^{\alpha+k}$ bad x's; thus, the probability that an x chosen uniformly at random from $\{0,1\}^{k_1}$ is bad is at most

$$2^{\alpha+k-k_1} \ = \ 2^{\frac{2k-k_1}{3}} \ \leq \ 2^{-k/3}$$

(using the fact that $k_1 \geq 3k$).

When x is good, the min-entropy of x — given (h, s) and $H_s(x)$ — is at least α since every $\tilde{x} \in N_x$ is equally likely. Let U_k represent the uniform distribution over $\{0,1\}^k$. Viewing h as a strong extractor (or, equivalently, applying the leftover-hash lemma [40]) we see that the statistical difference between $\{h, s, H_s(x), h(x)\}$ and $\{h, s, H_s(x), U_k\}$ is at most

$$2^{-(\alpha-k)/2} \ = \ 2^{\frac{2k-k_1}{3}} \ \leq \ 2^{-k/3}.$$

The claim, and hence the theorem, follows. ■

A practical setting of the above parameters (and one that we will use when discussing the efficiency of our scheme, below) is $k_1 = 448$, $k = 128$ which yields a 128-bit r with statistical difference at most $2 \cdot 2^{\frac{256-448}{3}} = 2^{-63}$ from uniform.[5] Also, in practice one would likely replace the UOWHF by a suitable modification of a cryptographic hash function such as SHA-1.

7.3 IBE Scheme

Boneh and Boyen [7] recently proposed two efficient IBE schemes satisfying the definition of security needed for our purposes. In the interests of space, we explore an instantiation of our construction using their first scheme only. (Of course, their second scheme could also be used. Doing so yields a mild efficiency improvement at the expense of requiring a stronger cryptographic assumption.)

We briefly discuss the cryptographic assumption on which the Boneh-Boyen IBE scheme is based. Let $\mathcal{IG}$ denote an efficient algorithm which, on input 1^k, outputs descriptions of two cyclic groups $\mathbb{G}, \mathbb{G}_1$ of prime order q (with $|q| = k$), a generator $g \in \mathbb{G}$, and an efficiently computable

[5]Note that since only second-preimage resistance is needed to achieve the binding property, a 128-bit output length provides sufficient security.

function $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ which is a *non-trivial bilinear map*; namely, (1) for all $\mu, \nu \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$ we have $\hat{e}(\mu^a, \nu^b) = \hat{e}(\mu, \nu)^{ab}$ and (2) $\hat{e}(g, g)$ is a generator of $\mathbb{G}_1$. (See [11] for a discussion about realizing an algorithm $\mathcal{IG}$ with these properties.) Following the standard terminology, we refer to $\hat{e}$ as a *pairing*.

The *computational bilinear Diffie-Hellman (BDH) problem with respect to* $\mathcal{IG}$ is the following: given $(\mathbb{G}, \mathbb{G}_1, g, \hat{e})$ as output by $\mathcal{IG}$ along with g^α, g^β, and g^γ (for random $\alpha, \beta, \gamma \in \mathbb{Z}_q$), compute $\hat{e}(g, g)^{\alpha\beta\gamma}$. Informally, we say that $\mathcal{IG}$ *satisfies the computational BDH assumption* if the computational BDH assumption with respect to $\mathcal{IG}$ is hard for any PPT algorithm.

The *decisional BDH problem with respect to* $\mathcal{IG}$ is to distinguish between tuples of the form $(g^\alpha, g^\beta, g^\gamma, \hat{e}(g, g)^{\alpha\beta\gamma})$ and $(g^\alpha, g^\beta, g^\gamma, \hat{e}(g, g)^\mu)$ for random $\alpha, \beta, \gamma, \mu \in \mathbb{Z}_q$ (note that $\hat{e}(g, g)^\mu$ is simply a random element of $\mathbb{G}_1$). Informally, we say $\mathcal{IG}$ *satisfies the decisional BDH assumption* if no PPT algorithm can solve the decisional BDH problem with respect to $\mathcal{IG}$ with probability significantly better than $\frac{1}{2}$. We refer to [11] for formal definitions and further discussion.

A concrete IBE scheme. We refer to [7] for the full details and content ourselves with giving only a high-level description of their first IBE scheme here, modified slightly for our eventual application. We assume for simplicity that system parameters $(\mathbb{G}, \mathbb{G}_1, g, \hat{e})$ have already been established by running $\mathcal{IG}(1^k)$ (of course, it is also possible for $\mathcal{IG}$ to be run during key generation). Let $G : \mathbb{G}_1 \to \{0, 1\}^k$ be a function whose output is indistinguishable from uniform when its input is uniformly distributed in $\mathbb{G}_1$ (however, G need not expand its input). The IBE scheme is defined as follows:

Setup Pick random generators $g_1, g_2 \in \mathbb{G}$ 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 the identity $ID \in \mathbb{Z}_q$ using $\mathsf{msk} = x$, choose a random $t \in \mathbb{Z}_q$ and return the key $SK_{ID} = (g_1^x g_2^t g_3^{t \cdot ID},\ g^t)$.

Encrypt To encrypt a message $M \in \{0, 1\}^k$ with respect to the identity $ID \in \mathbb{Z}_q$, choose a random $s \in \mathbb{Z}_q$ and output the ciphertext $(g^s,\ g_2^s g_3^{s \cdot ID},\ G(Z^s) \oplus M)$.

Decrypt To decrypt ciphertext (A, B, C) using private key (K_1, K_2), output:

$$C \oplus G(\hat{e}(A, K_1)/\hat{e}(B, K_2)). \tag{1}$$

Correctness can be easily verified. Security of the above scheme is based on the decisional[6] BDH assumption. For efficiency, the master secret key msk may also contain the discrete logarithms of g_1, g_2 (with respect to g), in which case the key-derivation algorithm requires only two exponentiations with respect to the fixed base g.

7.4 Putting it all Together

Given the above, we now fully describe a CCA-secure encryption scheme. In describing the scheme, we focus on the case of encrypting "long" messages (say, 10^4 bits or longer). Focusing on this case allows for a more accurate comparison with the scheme of [42] (which also focuses on this case).

[6] Note that if G instead represents a hard-core predicate for the *computational* BDH assumption, we obtain a scheme (encrypting a single bit) secure under this, possibly weaker, assumption. Running the scheme in parallel we obtain a scheme encrypting longer messages, as needed by our construction.

	Encryption	Decryption	Key generation	Ciphertext overhead
Our scheme	3.5 f-exps.	1.5 exp. + 1 pairing	4 f-exps.	$2 \cdot L_{\text{BG}} + 704$
KD-CS [42]	3.5 f-exps.	1.5 exps.	3 f-exps.	$2 \cdot L_{\text{DDH}} + 128$

Table 1: Efficiency comparison for CCA-secure encryption schemes. See text for discussion.

Let $(\mathsf{Mac}, \mathsf{Vrfy})$ denote the CBC-MAC using 128-bit AES as the underlying block cipher. Let $H : \{0,1\}^{448} \to \{0,1\}^{128}$ represent a hash function assumed to be second-preimage resistant (constructed, e.g., via a suitable modification of SHA-1). Let $G : \mathbb{G}_1 \to \{0,1\}^*$ denote a pseudorandom generator with sufficiently-long output length (constructed, e.g., by first hashing elements of $\mathbb{G}_1$ and then using a suitable modification of a block/stream cipher). We assume that $|q| > 128$ so that strings in $\{0,1\}^{128}$ may be mapped to $\mathbb{Z}_q$ in a one-to-one manner. Using the IBE scheme outlined above, we obtain the following (we assume that $G, H, q, \mathbb{G}, \mathbb{G}_1, \hat{e}, g$, and $\hat{e}(g,g)$ are provided as universal parameters):

Key generation Choose $\alpha_1, \alpha_2, x \leftarrow \mathbb{Z}_q$ and set $g_1 = g^{\alpha_1}$, $g_2 = g^{\alpha_2}$, and $g_3 = g^x$. Also set $Z = \hat{e}(g,g)^{\alpha_1 x}$. Finally, choose hash function h from a family of pairwise-independent hash functions. The public key is $PK = (g_1, g_2, g_3, Z, h)$ and the secret key is $SK = (\alpha_1, \alpha_2, x)$.

Encryption To encrypt message M using public key (g_1, g_2, g_3, Z, h), first choose random $r \in \{0,1\}^{448}$ and set $k_1 = h(r)$ and $ID = H(r)$. Choose random $s \in \mathbb{Z}_q$ and then set $C = (g^s, g_2^s g_3^{s \cdot ID}, G(Z^s) \oplus (M \circ r))$. Output the ciphertext

$$\langle ID, C, \mathsf{Mac}_{k_1}(C) \rangle.$$

Decryption To decrypt ciphertext $\langle ID, C, \mathsf{tag} \rangle$, first parse C as $(A, B, \hat{C})$. Then pick a random $t \in \mathbb{Z}_q$ and compute the values $(M \circ r) = \hat{C} \oplus G(\hat{e}(A^{\alpha_1 x + t(\alpha_2 + x \cdot ID)} B^{-t}, g))$. Set $k_1 = h(r)$. If $\mathsf{Vrfy}_{k_1}(C, \mathsf{tag}) \stackrel{?}{=} 1$ and $H(r) \stackrel{?}{=} ID$ output M; otherwise, output $\perp$.

We have changed the steps used in decryption for efficiency purposes, but it is easily checked that decryption yields the same result as deriving a secret key for identity ID and then using this key to decrypt C.

We tabulate the efficiency of our scheme, and compare it to the Kurosawa-Desmedt variant of Cramer-Shoup encryption [42, 21] (which we refer to as KD-CS), in Table 1. In tabulating computational efficiency, "private-key" operations (that is, evaluations of G, H, and h) and group multiplications are ignored; "exp" stands for exponentiation; "f-exp" refers to exponentiation relative to a fixed base (where efficiency can be improved using pre-computation); and one multi-exponentiation is counted as 1.5 exponentiations [46, p. 618]. Ciphertext overhead is the difference (in bits) between the lengths of the ciphertext and the message. L_{BG} is the bit-length of an element in a group $\mathbb{G}$ suitable for our scheme, and L_{DDH} is the bit-length of an element in a group suitable for the KD-CS scheme.

Although performance of the two systems looks similar, efficiency of the KD-CS scheme scales better with the security parameter. To see why, fix the measure of hardness to be the difficulty of computing discrete logarithms in the respective groups. (Although the underlying computational problems used to prove security of the above two schemes are not known to be equivalent to the discrete logarithm problem, in each case the best currently-known algorithms for solving the problem rely on a discrete logarithm computation.) Consider two concrete settings:

80-bit security. Suppose we wish to use groups in which solving the discrete logarithm problem (using the best currently-known algorithms) is roughly equivalent to the security attained by 80-bit symmetric-key cryptography.

- Our scheme can use groups based on so-called MNT elliptic curves [48]. In this case, the discrete logarithm problem in a group $\mathbb{G}$ in which elements can be written using $L_{\mathrm{BG}} = \log q$ bits (q prime) can be reduced to a discrete logarithm problem in $\mathbb{F}^*_{q^6}$. (See [13, Section 4.3].) 80-bit security for the latter is obtained by setting $q^6 \approx 2^{1024}$ [44, 1] (specifically, our numbers throughout this discussion are taken from [1, Table 2]). We thus need $L_{\mathrm{BG}} \approx 1024/6 \approx 171$.
- The KD-CS scheme can use standard elliptic curve groups, for which the best-known algorithm for computing discrete logarithms is Pollard's rho algorithm that runs in time proportional to $\sqrt{q}$ for groups of order q. This gives $L_{\mathrm{DDH}} = \log q \approx 160$ [1].

We see that for this level of security, both schemes have ciphertexts of roughly the same length and group operations take roughly the same amount of time. Still, the KD-CS scheme outperforms our scheme (even if by a relatively small margin in some cases) in all parameters; this is especially true for decryption since a pairing calculation is computationally expensive compared to an exponentiation.

256-bit security. Now, the KD-CS scheme performs even better relative to ours.

- For our scheme, we can use groups based on a certain class of elliptic curves suggested by Barreto and Naehrig [4]. (Note that such curves will give worse performance for the case of 80-bit security considered above.) Now, the discrete logarithm problem in a group $\mathbb{G}$ in which elements can be written using $L_{\mathrm{BG}} = \log q$ bits (q prime) can be reduced to a discrete logarithm problem in $\mathbb{F}^*_{q^{12}}$. 256-bit security for the latter is obtained by setting $q^{12} \approx 2^{15360}$. We thus need $L_{\mathrm{BG}} \approx 15386/12 \approx 1280$.
- Using the same the analysis as before, the KD-CS scheme can use $L_{\mathrm{DDH}} = \log q \approx 512$.

We conclude that for currently acceptable settings of the security parameter the schemes have comparable performance, though the KD-CS scheme is more efficient. We stress that our goal is not to displace the KD-CS system but rather to show another approach to building practical CCA-secure systems. Our construction also has advantages not present in the KD-CS scheme, such as being readily amenable to a threshold implementation [10].

8 Conclusions

We presented in this paper new paradigms for constructing CCA-secure public-key encryption schemes using IBE as a building block. Our paradigms extend to enable constructions of CCA-secure (hierarchical) identity-based encryption schemes as well. Instantiating our constructions with an existing IBE system yields a CCA-secure encryption scheme whose performance, for standard settings of the security parameter, is competitive with the best CCA-secure schemes known to date.

Acknowledgments

We thank Eu-Jin Goh for pointing out that our techniques imply a conversion from IBE to non-adaptive CCA security with essentially no overhead. We also thank the anonymous referees for their helpful comments, and in particular for suggesting a way to simplify the presentation of the proof of Theorem 2

References

[1] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for Key Management — Part 1: General. NIST Special Publication 800-57, August 2005, National Institute of Standards and Technology. Available at `http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf`

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. *Adv. in Cryptology — Crypto '98*, LNCS vol. 1462, Springer-Verlag, pp. 26–45, 1998.

[3] M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. *1st ACM Conf. on Computer and Communications Security*, ACM, pp. 62–73, 1993.

[4] P. Barreto and M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. *Selected Areas in Cryptography — SAC 2005*, LNCS vol. 3897, Springer-Verlag, pp. 319–331, 2006.

[5] D. Bleichenbacher. Chosen-Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. *Adv. in Cryptology — Crypto '98*, LNCS vol. 1462, Springer-Verlag, pp. 1–12, 1998.

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

[7] 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 at `http://eprint.iacr.org/2004/172`

[8] 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. Full version available at `http://eprint.iacr.org/2004/173`

[9] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical Identity-Based Encryption With Constant-Size Ciphertexts. *Adv. in Cryptology — Eurocrypt 2005*, LNCS vol. 3494, Springer-Verlag, pp. 440–456, 2005. Full version available at `http://eprint.iacr.org/2005/015`

[10] D. Boneh, X. Boyen, and S. Halevi. Chosen-Ciphertext Secure Public-Key Threshold Encryption Without Random Oracles. *Topics in Cryptology — CT-RSA 2006*, LNCS vol. 3860, Springer-Verlag, pp. 226–243, 2006.

[11] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Computing* 32(3): 586–615 (2003).

[12] D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. *Topics in Cryptology — CT-RSA 2005*, LNCS vol. 3376, Springer-Verlag, pp. 87–103, 2005.

[13] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *J. Cryptology* 17(4): 297–319 (2004).

[14] X. Boyen, Q. Mei, and B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. *12th ACM Conference on Computer and Communications Security*, ACM, pp. 320–329, 2005.

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

[16] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. *J. ACM* 51(4): 557–594 (2004).

[17] 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` and to appear in *J. Cryptology*.

[18] 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.

[19] R. Canetti, H. Krawczyk, and J.B. Nielsen. Relaxing Chosen Ciphertext Security. *Adv. in Cryptology — Crypto 2003*, LNCS vol. 2656, Springer-Verlag, pp. 565–582, 2003.

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

[21] R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen-Ciphertext Attack. *SIAM J. Computing* 33(1): 167–226 (2003).

[22] 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.

[23] 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.

[24] 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 '93*, LNCS vol. 773, Springer-Verlag, pp. 250–265, 1993.

[25] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. *Adv. in Cryptology — Crypto '89*, LNCS vol. 435, Springer-Verlag, pp. 307–315, 1990.

[26] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-Interactive and Non-Malleable Commitment. *30th ACM Symposium on Theory of Computing*, ACM, pp. 141–150, 1998.

[27] G. Di Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. Efficient and Non-Interactive Non-Malleable Commitment. *Adv. in Cryptology — Eurocrypt 2001*, LNCS vol. 2045, Springer-Verlag, pp. 40–59, 2001.

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

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

[30] S. Even, O. Goldreich, and S. Micali. On-Line/Off-Line Digital Signatures. *J. Cryptology* 9(1): 35–67 (1996).

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

[32] 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.

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

[34] C. Gentry. Practical Identity-Based Encryption Without Random Oracles. *Adv. in Cryptology — Eurocrypt 2006*, LNCS vol. 4004, Springer-Verlag, pp. 445–464, 2006.

[35] O. Goldreich. A Uniform Complexity Treatment of Encryption and Zero-Knowledge. *J. Cryptology* 6(1): 21–53 (1993).

[36] O. Goldreich. *Foundations of Cryptography, vol. 2: Basic Applications*. Cambridge University Press, 2004.

[37] O. Goldreich, Y. Lustig, and M. Naor. On Chosen-Ciphertext Security of Multiple Encryptions. Available at `http://eprint.iacr.org/2002/089`

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

[39] S. Halevi and S. Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. *Adv. in Cryptology — Crypto '96*, LNCS vol. 1109, Springer-Verlag, pp. 201–215, 1996.

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

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

[42] 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.

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

[44] A.K. Lenstra and E.R. Verheul. Selecting Cryptographic Key Sizes. *J. Cryptology* 14(4): 255–293 (2001).

[45] P. MacKenzie, M. Reiter, and K. Yang. Alternatives to Non-Malleability: Definitions, Constructions, and Applications. *1st Theory of Cryptography Conference — TCC 2004*, LNCS vol. 2951, Springer-Verlag, pp. 171–190, 2004.

[46] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1997.

[47] S. Micali, C. Rackoff, and B. Sloan. The Notion of Security for Probabilistic Cryptosystems. *SIAM J. Computing* 17(2): 412–426 (1988).

[48] A. Miyaji, M. Nakabayashi, and S. Takano. New Explicit Constructions of Elliptic Curve Traces for FR-Reduction. *IEICE Trans. Fundamentals* E84-A(5): 1234–1243 (2001).

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

[50] C. Rackoff and D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *Adv. in Cryptology — Crypto '91*, LNCS vol. 576, Springer-Verlag, pp. 433–444, 1992.

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

[52] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. *Adv. in Cryptology — Crypto '84*, LNCS vol. 196, Springer-Verlag, pp. 47–53, 1985.

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

[54] D.R. Stinson. Universal Hashing and Authentication Codes. *Designs, Codes, and Cryptography* 4(4): 369–380 (1994).

[55] Y. Watanabe, J. Shikata, and H. Imai. Equivalence Between Semantic Security and Indistinguishability Against Chosen-Ciphertext Attacks. *Public-Key Cryptography 2003*, LNCS vol. 2567, Springer-Verlag, pp. 71–84, 2003.

[56] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. *Adv. in Cryptology — Eurocrypt 2005*, LNCS vol. 3494, Springer-Verlag, pp. 114–127, 2005.

[57] 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).

A Review of Standard Definitions

We provide the standard definitions of public-key encryption schemes and their security against adaptive chosen-ciphertext attacks, as well as appropriate definitions for strong one-time signature schemes and message authentication codes.

A.1 Public-Key Encryption

Definition 7 A *public-key encryption scheme* 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.
- The randomized *encryption algorithm* $\mathcal{E}$ takes as input a public key PK and a message m (in some implicit message space), and outputs a ciphertext C. We write $C \leftarrow \mathcal{E}_{PK}(m)$.
- The (possibly randomized) *decryption algorithm* $\mathcal{D}$ takes as input a ciphertext C and a secret key SK. It returns a message m or the symbol $\perp$ (which is not in the message space). We write $m \leftarrow \mathcal{D}_{SK}(C)$.

We require that for all (PK, SK) output by Gen, all m in the message space, 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. [2]).

Definition 8 A public-key encryption scheme Π is *secure against adaptive chosen-ciphertext attacks* (i.e., "CCA-secure") if the advantage of any PPT adversary $\mathcal{A}$ in the following game is negligible in the security parameter k:

1. $\mathsf{Gen}(1^k)$ outputs (PK, SK). Adversary $\mathcal{A}$ 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, $\mathcal{A}$ 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. $\mathcal{A}$ may continue to query its decryption oracle $\mathcal{D}_{SK}(\cdot)$ except that it may not request the decryption of C^*.
5. Finally, $\mathcal{A}$ outputs a guess b'.

We say that $\mathcal{A}$ *succeeds* if $b' = b$, and denote the probability of this event by $\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ}]$. The adversary's *advantage* is defined as $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A},\Pi}(k) \stackrel{\text{def}}{=} \left|\Pr^{\mathsf{PKE}}_{\mathcal{A},\Pi}[\mathsf{Succ}] - 1/2\right|$. ♢

A.2 Signatures and MACs

We remind the reader of the standard functional definitions for signature schemes and message authentication codes, followed by a definition of strong one-time security appropriate for each.

Definition 9 A *signature scheme* is a triple of PPT algorithms $(\mathcal{G}, \mathsf{Sign}, \mathsf{Vrfy})$ such that:

- The randomized *key generation algorithm* $\mathcal{G}$ takes as input the security parameter 1^k and outputs a verification key vk and a signing key sk. We assume for simplicity that the length of vk is fixed for any given value of k.
- The *signing algorithm* Sign takes as input a signing key sk and a message m (in some implicit message space), and outputs a signature σ. We write $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.
- The *verification algorithm* Vrfy takes as input a verification key vk, a message m, and a signature σ, and outputs a bit $b \in \{0,1\}$ (where $b = 1$ signifies "acceptance" and $b = 0$ signifies "rejection"). We write this as $b := \mathsf{Vrfy}_{vk}(m, \sigma)$.

We require that for all (vk, sk) output by $\mathcal{G}$, all m in the message space, and all σ output by $\mathsf{Sign}_{sk}(m)$, we have $\mathsf{Vrfy}_{vk}(m, \sigma) = 1$. ♢

A message authentication code is similar in spirit to a signature scheme, except that here the signing key and verification key are identical. We review the definition for convenience:

Definition 10 A *message authentication code* is a pair of PPT algorithms (Mac, Vrfy) such that:

- The *tagging algorithm* Mac takes as input a key $sk \in \{0,1\}^k$ (where k is the security parameter) and a message m (in some implicit message space). It outputs a tag tag, and we denote this by $\mathsf{tag} \leftarrow \mathsf{Mac}_{sk}(m)$.
- The *verification algorithm* Vrfy takes as input a key sk, a message m, and a tag tag; it outputs a bit $b \in \{0,1\}$ (where $b = 1$ signifies "acceptance" and $b = 0$ signifies "rejection"). We write this as $b := \mathsf{Vrfy}_{sk}(m, \mathsf{tag})$.

We require that for all sk, all m in the message space, and all tag output by $\mathsf{Mac}_{sk}(m)$, we have $\mathsf{Vrfy}_{sk}(m, \mathsf{tag}) = 1$. ♢

We next turn to definitions of security for signature schemes and message authentication codes. The definition of security is analogous in each case: the adversary should be unable to forge a valid message/signature (resp., message/tag) pair, after receiving a signature (resp., tag) on any *single* message m of the adversary's choice. Note that we require so-called *strong* security in each case, so that it should be infeasible for the adversary to generate even a different signature (resp., tag) on the same message m.

Definition 11 A signature scheme Sig is a *strong one-time signature scheme* if the success probability of any PPT adversary $\mathcal{A}$ in the following game is negligible in the security parameter k:

1. $\mathcal{G}(1^k)$ outputs (vk, sk) and the adversary is given 1^k and vk.
2. $\mathcal{A}(1^k, vk)$ may do one of the following:
   (a) $\mathcal{A}$ may output a pair (m^*, σ^*) and halt. In this case (m, σ) are undefined.
   (b) $\mathcal{A}$ may output a message m, and is then given in return $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$. Following this, $\mathcal{A}$ outputs (m^*, σ^*).

We say the adversary *succeeds* if $\mathsf{Vrfy}_{vk}(m^*, \sigma^*) = 1$ but $(m^*, \sigma^*) \neq (m, \sigma)$ (assuming (m, σ) are defined). We stress that the adversary may succeed even if $m^* = m$. ♢

Definition 12 A message authentication code (Mac, Vrfy) is a *strong one-time message authentication code* if the success probability of any PPT adversary $\mathcal{A}$ in the following game is negligible in the security parameter k:

1. A random key $sk \in \{0,1\}^k$ is chosen.
2. $\mathcal{A}(1^k)$ may do one of the following:
   (a) $\mathcal{A}$ may output (m^*, tag^*). In this case, (m, tag) are undefined.
   (b) $\mathcal{A}$ may output a message m and is then given in return $\mathsf{tag} \leftarrow \mathsf{Mac}_{sk}(m)$. Following this, $\mathcal{A}$ outputs (m^*, tag^*).

We say the adversary *succeeds* if $\mathsf{Vrfy}_{sk}(m^*, \mathsf{tag}^*) = 1$ but $(m^*, \mathsf{tag}^*) \neq (m, \mathsf{tag})$ (assuming (m, tag) are defined). We stress that the adversary may succeed even if $m^* = m$. ♢