On the Joint Security of Encryption and Signature, Revisited

Kenneth G. Paterson[1⋆], Jacob C.N. Schuldt[2⋆⋆], Martijn Stam[3], and Susan Thomson[1⋆⋆⋆]

[1] Royal Holloway, University of London
[2] Research Center for Information Security, AIST, Japan
[3] University of Bristol

Abstract. We revisit the topic of joint security for combined public key schemes, wherein a single keypair is used for both encryption and signature primitives in a secure manner. While breaking the principle of key separation, such schemes have attractive properties and are sometimes used in practice. We give a general construction for a combined public key scheme having joint security that uses IBE as a component and that works in the standard model. We provide a more efficient direct construction, also in the standard model.

1 Introduction

Key separation versus key reuse: The folklore principle of key separation dictates using different keys for different cryptographic operations. While this is well-motivated by real-world, security engineering concerns, there are still situations where it is desirable to use the same key for multiple operations [15]. In the context of public key cryptography, using the same keypair for both encryption and signature primitives can reduce storage requirements (for certificates as well as keys), reduce the cost of key certification and the time taken to verify certificates, and reduce the footprint of cryptographic code. These savings may be critical in embedded systems and low-end smart card applications. As a prime example, the globally-deployed EMV standard for authenticating credit and debit card transactions allows the same keypair to be reused for encryption and signatures for precisely these reasons [11].

However, this approach of reusing keys is not without its problems. For example, there is the issue that encryption and signature keypairs may have different lifetimes, or that the private keys may require different levels of protection [15]. Most importantly of all, there is the question of whether it is *secure* to use the same keypair in two (or more) different primitives – perhaps the two uses will interact with one another badly, in such a way as to undermine the security of

⋆ This author is supported by EPSRC Leadership Fellowship EP/H005455/1.
⋆⋆ This author is supported by a JSPS Fellowship for Young Scientists.
⋆⋆⋆ This author is supported by the EPSRC through Leadership Fellowship EP/H005455/1.

one or both of the primitives. In the case of textbook RSA, it is obvious that using the same keypair for decryption and signing is dangerous, since the signing and decryption functions are so closely related in this case. Security issues may still arise even if some standardized padding is used prior to encryption and signing [20]. In Section 3 we will provide another example in the context of encryption and signature primitives, where the individual components are secure (according to the usual notions of security for encryption and signature) but become completely insecure as soon as they are used in combination with one another. At the protocol level, Kelsey, Schneier and Wagner [18] gave examples of protocols that are individually secure, but that interact badly when a keypair is shared between them.

The formal study of the security of key reuse was initiated by Haber and Pinkas [15]. They introduced the concept of a *combined public key scheme*. Here, an encryption scheme and signature scheme are combined: the existing algorithms to encrypt, decrypt, sign and verify are preserved, but the two key generation algorithms are modified to produce a single algorithm. This algorithm outputs two keypairs, one for the encryption scheme and one for the signature scheme, with the keypairs no longer necessarily being independent. Indeed, under certain conditions, the two keypairs may be identical, in which case the savings described above may be realised. In other cases, the keypairs are not identical but can have some shared components, leading to more modest savings. Haber and Pinkas also introduced the natural security model for combined public key schemes, where the adversary against the encryption part of the scheme is equipped with a signature oracle in addition to the usual decryption oracle, and where the adversary against the signature part of the scheme is given a decryption oracle in addition to the usual signature oracle. In this setting, we talk about the *joint security* of the combined scheme.

Setting a benchmark: As we shall see in Section 3, there is a trivial "Cartesian product" construction for a combined public key scheme with joint security. The construction uses arbitrary encryption and signature schemes as components, and the combined scheme's keypair is just a pair of vectors whose components are the public/private keys of the component schemes. Thus the Cartesian product construction merely formalises the principle of key separation. This construction, while extremely simple, provides a benchmark by which other constructions can be judged. For example, if the objective is to minimise the public key size in a combined scheme, then any construction should aim to have shorter keys than can be obtained by instantiating the Cartesian product construction with the best available encryption and signature schemes.

Re-evaluating Haber-Pinkas: In this respect, we note that, while Haber and Pinkas considered various well-known concrete schemes and conditions under which their keys could be partially shared, none of their examples having provable security in the standard model lead to *identical* keypairs for both signature and encryption. Indeed, while the approach of Haber and Pinkas can be made to work in the random oracle model by careful oracle programming and domain sep-

aration, their approach does not naturally extend to the standard model. More specifically, in their approach, to be able to simulate the signing oracle in the IND-CCA security game, the public key of the combined scheme cannot be exactly the same as the public key of the underlying encryption scheme (otherwise, successful simulation would lead to a signature forgery). This makes it hard to achieve full effective overlap between the public keys for signing and encryption. For the (standard model) schemes considered by Haber and Pinkas this results in the requirements that part of the public key be specific to the encryption scheme and that another part of it be specific to the signature scheme. Furthermore, at the time of publication of [15] only a few secure (IND-CCA2, resp. EUF-CMA) and efficient standard-model schemes were known. Consequently, no "compatible" signature and encryption schemes were identified in [15] for the standard model.

Combined schemes from trapdoor permutations: The special case of combined schemes built from trapdoor permutations was considered in [8, 21]. Here, both sets of authors considered the use of various message padding schemes in conjunction with an arbitrary trapdoor permutation to build combined public key schemes having joint security. Specifically, Coron et al. [8] considered the case of PSS-R encoding, while Komano and Ohta [21] considered the cases of OAEP+ and REACT encodings. All of the results in these two papers are in the random oracle model. In further related, but distinct, work, Dodis et al. [10] (see also [9]) considered the use of message padding schemes and trapdoor permutations to build signcryption schemes. Dodis et al. showed, again in the random oracle model, how to build efficient, secure signcryption schemes in which each user's keypair, specifying a permutation and its trapdoor, is used for both signing and encryption purposes.

1.1 Our Contribution

We focus on the problem of how to construct combined public key schemes which are jointly secure in the standard model, a problem for which, as we have explained above, there currently exist no fully satisfactory solutions. Naturally, for reasons of practical efficiency, we are interested in minimising the size of keys (both public and private), ciphertexts, and signatures in such schemes. The complexity of the various algorithms needed to implement the schemes will also be an important consideration.

As a warm-up, in Section 3, we give the simple Cartesian product construction, as well as a construction showing that the general problem is not vacuous (i.e. that there exist insecure combined schemes whose component schemes are secure when used in isolation).

We then present in Section 4 a construction for a combined public key scheme using an IBE scheme as a component. The trick here is to use the IBE scheme in the Naor transform and the CHK transform *simultaneously* to create a combined public key scheme that is jointly secure, under rather weak requirements on the starting IBE scheme (specifically, the IBE scheme needs to be OW-ID-CPA

and IND-sID-CPA secure). This construction extends easily to the (hierarchical) identity-based setting. Instantiating this construction using standard model secure IBE schemes from the literature already yields rather efficient combined schemes. For example, using an asymmetric pairing version of Gentry's IBE scheme [14], we can achieve a combined scheme in which, at the 128-bit security level, the public key size is 1536 bits, the signature size is 768 bits and the ciphertext size is 2304 bits (plus the size of a signature and a verification key for a one-time signature scheme), with joint security being based on a q-type assumption. This is already competitive with schemes arising from the Cartesian product construction.

We then provide a more efficient direct construction for a combined scheme with joint security in Section 5. This construction is based on the signature scheme of Boneh and Boyen [4] and a KEM obtained by applying the techniques by Boyen, Mei and Waters [7] to the second IBE scheme of Boneh and Boyen in [3]. At the 128-bit security level, it enjoys public keys that consist of 1280 bits, signatures that are 768 bits and a ciphertext overhead of just 512 bits. The signatures can be shrunk at the cost of increasing the public key size.

The ideas of this paper also have applications for signcryption. We show in the full version [24] that a (tag-based) combined public key scheme can be used to construct a signcryption scheme, using the "sign-then-encrypt" construction of [23], that is secure in the strongest security model for signcryption (achieving insider confidentiality and insider unforgeability in the multi-user setting). Instantiating this construction with our concrete combined public key scheme effectively solves the challenge implicitly laid down by Dodis et al. in [9], to construct an efficient standard model signcryption scheme in which a single short keypair can securely be used for both sender and receiver functions. Furthermore, we are able to show that the signcryption scheme we obtain is jointly secure when used in combination with *both* its signature and encryption components. Thus we are able to obtain a triple of functionalities (signcryption, signature, encryption) which are jointly secure using only a single keypair.

1.2 Further Related Work

Further work on combined public key schemes in the random oracle model, for both the normal public key setting and the identity-based setting can be found in [27]. In particular, it is proved that the identity-based signature scheme of Hess [16] and Boneh and Franklin's identity-based encryption scheme [6] can be used safely together.

The topic of joint security of combined public key schemes is somewhat linked to the topic of cryptographic agility [1], which considers security when the same key (or key pair) is used simultaneously in multiple instantiations of the *same* cryptographic primitive. This contrasts with joint security, where we are concerned with security when the same key pair is used simultaneously in instantiations of *different* cryptographic primitives. The connections between these different but evidently related topics remain to be explored.

2 Preliminaries

In our constructions, we will make use of a number of standard primitives, including digital signatures, (tag-based) public key encryption, identity-based encryption (IBE), a data encapsulation mechanism (DEM), and an always second-preimage resistant hash function. We refer the reader to the full version [24] for the standard definitions and security notions for these primitives. In the following, we briefly recall the properties of bilinear pairings as well as define the computational assumptions which we will make use of to prove the security of our concrete constructions.

Bilinear pairings: Let $\mathbb{G}_1 = \langle g_1 \rangle$, $\mathbb{G}_2 = \langle g_2 \rangle$, $\mathbb{G}_T$ be groups of prime order p. A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies the following properties:

1. Bilinear: For all $a, b \in \mathbb{Z}, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
2. Non-degenerate: $e(g_1, g_2) \neq 1$.
3. Computable: There is an efficient algorithm to compute the map e.

Note that we work exclusively in the setting of asymmetric pairings, whereas schemes are often presented in the naive setting of symmetric pairings $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. At higher security levels (128 bits and above), asymmetric pairings are far more efficient both in terms of computation and in terms of the size of group elements [13]. As a concrete example, using BN curves [2] and sextic twists, we can attain the 128-bit security level with elements of $\mathbb{G}_1$ being represented by 256 bits and elements of $\mathbb{G}_2$ needing 512 bits. By exploiting compression techniques [26], elements of $\mathbb{G}_T$ in this case can be represented using 1024 bits. For further details on parameter selection for pairings, see [12].

Strong Diffie-Hellman (SDH) assumption [4]: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p, respectively generated by g_1 and g_2. In the bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$, the q-SDH problem is stated as follows:

> Given as input a $(q+3)$-tuple of elements
> $\left(g_1, g_1^x, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)}\right) \in \mathbb{G}_1^2 \times \mathbb{G}_2^{q+1}$
> output a pair $\left(c, g_2^{1/(x+c)}\right) \in \mathbb{Z}_p \times \mathbb{G}_2$ for a freely chosen value $c \in \mathbb{Z}_p \backslash \{-x\}$.

An algorithm $\mathcal{A}$ solves the q-SDH problem in the bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with advantage ϵ if

$$\Pr\left[\mathcal{A}\left(g_1, g_1^x, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)}\right) = \left(c, g_2^{1/(x+c)}\right)\right] \geq \epsilon,$$

where the probability is over the random choice of generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, the random choice of $x \in \mathbb{Z}_p^*$, and the random bits consumed by $\mathcal{A}$. We say that the (t, q, ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$.

Decisional Bilinear Diffie-Hellman Inversion (DBDHI) assumption [3]: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p, respectively generated by g_1 and g_2. In the bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$, the q-DBDHI problem is stated as follows:

> Given as input a $(q+4)$-tuple of elements
> $\left(g_1, g_1^x, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)}, T\right) \in \mathbb{G}_1^2 \times \mathbb{G}_2^{q+1} \times \mathbb{G}_T$
> output 0 if $T = e(g_1, g_2)^{1/x}$ or 1 if T is a random element in $\mathbb{G}_T$.

An algorithm $\mathcal{A}$ solves the q-DBDHI problem in the bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with advantage ϵ if

$$\left| \Pr\left[\mathcal{A}\left(g_1, g_1^x, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)}, e(g_1, g_2)^{1/x}\right) = 0\right] - \Pr\left[\mathcal{A}\left(g_1, g_1^x, g_2, g_2^x, g_2^{(x^2)}, \ldots, g_2^{(x^q)}, T\right) = 0\right] \right| \geq \epsilon,$$

where the probability is over the random choice of generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, the random choice of $x \in \mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{A}$. We say that the (t, q, ϵ)-DBDHI assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has advantage at least ϵ in solving the q-DBDHI problem in $(\mathbb{G}_1, \mathbb{G}_2)$.

3 Combined Signature and Encryption Schemes

A combined signature and encryption scheme is a combination of a signature scheme and a public key encryption scheme that share a key generation algorithm and hence a keypair (pk, sk). It comprises a tuple of algorithms (KeyGen, Sign, Verify, Encrypt, Decrypt) such that (KeyGen, Sign, Verify) form a signature scheme and (KeyGen, Encrypt, Decrypt) form a PKE scheme. Since the signature and PKE schemes share a keypair the standard notions of EUF-CMA and IND-CCA security need to be extended to reflect an adversary's ability to request both signatures and decryptions under the challenge public key. When defining a security game against a component of the scheme the nature of any additional oracles depends on the required security of the other components. For example, if EUF-CMA security of the signature component of a combined signature and encryption scheme is required, then it is necessary to provide the adversary with unrestricted access to a signature oracle when proving IND-CCA security of the encryption component of the scheme. The security definitions given implicitly in [8], considering IND-CCA security of the encryption component and EUF-CMA security of the signature component, are stated formally here.

EUF-CMA security in the presence of a decryption oracle: Let (KeyGen, Sign, Verify, Encrypt, Decrypt) be a combined signature and encryption scheme. Existential unforgeability of the signature component under an adaptive chosen message attack in the presence of an additional decryption oracle is defined through the following game between a challenger and an adversary $\mathcal{A}$.

Setup: The challenger generates a keypair $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$ and gives $\mathcal{A}$ the challenge public key pk.

Query phase: $\mathcal{A}$ requests signatures on messages m_i of its choice. The challenger responds to each signature query with a signature $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$. $\mathcal{A}$ also requests decryptions of ciphertexts c_i of its choice. The challenger responds to each decryption query with a message $m \leftarrow \mathsf{Decrypt}(sk, c_i)$ or a failure symbol $\perp$.

Forgery: $\mathcal{A}$ outputs a message signature pair (σ, m) such that m was not submitted to the signing oracle, and wins the game if $\mathsf{Verify}(pk, \sigma, m) = 1$.

The advantage of an adversary $\mathcal{A}$ is the probability it wins the above game.

A forger $\mathcal{A}$ (t, q_d, q_s, ϵ)-breaks the signature component of a combined signature and encryption scheme if $\mathcal{A}$ runs in time at most t, makes at most q_d decryption queries and q_s signature queries and has advantage at least ϵ. The signature component of a combined signature and encryption scheme is said to be (t, q_d, q_s, ϵ)-EUF-CMA secure in the presence of a decryption oracle if no forger (t, q_d, q_s, ϵ)-breaks it.

IND-CCA security in the presence of a signing oracle: Let $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a combined signature and encryption scheme. Indistinguishability of the encryption component under an adaptive chosen ciphertext attack in the presence of an additional signing oracle is defined through the following game between a challenger and an adversary $\mathcal{A}$.

Setup: The challenger generates a keypair $(pk, sk) \leftarrow \mathsf{Keyen}(1^k)$ and gives $\mathcal{A}$ the challenge public key pk.

Phase 1: $\mathcal{A}$ requests decryptions of ciphertexts c_i of its choice. The challenger responds to each decryption query with a message $m \leftarrow \mathsf{Decrypt}(sk, c_i)$ or a failure symbol $\perp$. $\mathcal{A}$ also requests signatures on messages m_i of its choice. The challenger responds to each signature query with a signature $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$.

Challenge: $\mathcal{A}$ chooses two equal length messages m_0, m_1. The challenger chooses a random bit b, computes $c^* \leftarrow \mathsf{Encrypt}(pk, m_b)$, and passes c^* to the adversary.

Phase 2: As Phase 1 but with the restriction that $\mathcal{A}$ must not request the decryption of the challenge ciphertext c^*.

Guess: $\mathcal{A}$ outputs a guess b' for b.

The advantage of $\mathcal{A}$ is $\left|\Pr[b' = b] - \frac{1}{2}\right|$.

An adversary $\mathcal{A}$ (t, q_d, q_s, ϵ)-breaks the encryption component of a combined signature and encryption scheme if $\mathcal{A}$ runs in time at most t, makes at most q_d decryption queries and q_s signature queries and has advantage at least ϵ. The encryption component of a combined signature and encryption scheme is said to be (t, q_d, q_s, ϵ)-IND-CCA secure in the presence of a signing oracle if no adversary (t, q_d, q_s, ϵ)-breaks it.

Informally, we say that a combined scheme is *jointly secure* if it is both EUF-CMA secure in the presence of a decryption oracle and IND-CCA secure in the presence of a signing oracle.

3.1 A Cartesian Product Construction

A trivial way of obtaining a system satisfying the above security properties is to concatenate the keys of an encryption scheme and signature scheme, then use the appropriate component of the compound key for each operation. This gives a combined signature and encryption scheme where the signature and encryption operations are essentially independent. Consequently their respective security properties are retained in the presence of the additional oracle. This simple construction sets a benchmark in terms of key size and other performance measures that any bespoke construction should best in one or more metrics.

Formally, let $\mathcal{S} = (\mathcal{S}.\mathsf{KeyGen}, \mathcal{S}.\mathsf{Sign}, \mathcal{S}.\mathsf{Verify})$ be a signature scheme, and let $\mathcal{E} = (\mathcal{E}.\mathsf{KeyGen}, \mathcal{E}.\mathsf{Encrypt}, \mathcal{E}.\mathsf{Decrypt})$ be an encryption scheme. Then the Cartesian product combined signature and encryption scheme $\mathsf{CartCSE}(\mathcal{E}, \mathcal{S})$ is constructed as follows:

$\mathsf{CartCSE}(\mathcal{E}, \mathcal{S}).\mathsf{KeyGen}(1^k)$**:** Run $\mathcal{S}.\mathsf{KeyGen}(1^k)$ to get (pk_s, sk_s). Run $\mathcal{E}.\mathsf{KeyGen}(1^k)$ to get (pk_e, sk_e). Output the public key $pk = (pk_s, pk_e)$ and the private key $sk = (sk_s, sk_e)$.
$\mathsf{CartCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Sign}(sk, m)$**:** Output $\mathcal{S}.\mathsf{Sign}(sk_s, m)$.
$\mathsf{CartCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Verify}(pk, \sigma, m)$**:** Output $\mathcal{S}.\mathsf{Verify}(pk_s, \sigma, m)$.
$\mathsf{CartCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Encrypt}(pk, m)$**:** Output $\mathcal{E}.\mathsf{Encrypt}(pk_e, m)$.
$\mathsf{CartCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Decrypt}(sk, c)$**:** Output $\mathcal{E}.\mathsf{Decrypt}(sk_e, c)$.

We omit the straightforward proof that this scheme is jointly secure if $\mathcal{S}$ is EUF-CMA secure and $\mathcal{E}$ is IND-CCA secure.

3.2 An Insecure CSE Scheme whose Components are Secure

To show that the definitions are not trivially satisfied, we give a pathological example to show that a PKE scheme and a signature scheme that are individually secure may not be secure when used in combination. Let $\mathcal{S} = (\mathcal{S}.\mathsf{KeyGen}, \mathcal{S}.\mathsf{Sign}, \mathcal{S}.\mathsf{Verify})$ be an EUF-CMA secure signature scheme, and let $\mathcal{E} = (\mathcal{E}.\mathsf{KeyGen}, \mathcal{E}.\mathsf{Encrypt}, \mathcal{E}.\mathsf{Decrypt})$ be an IND-CCA secure encryption scheme. A combined signature and encryption scheme $\mathsf{BadCSE}(\mathcal{E}, \mathcal{S})$ can be constructed as follows.

$\mathsf{BadCSE}(\mathcal{E}, \mathcal{S}).\mathsf{KeyGen}(1^k)$**:** Run $\mathcal{S}.\mathsf{KeyGen}(1^k)$ to get (pk_s, sk_s). Run $\mathcal{E}.\mathsf{KeyGen}(1^k)$ to get (pk_e, sk_e). Output the public key $pk = (pk_s, pk_e)$ and the private key $sk = (sk_s, sk_e)$.
$\mathsf{BadCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Sign}(sk, m)$**:** Compute $\sigma' = \mathcal{S}.\mathsf{Sign}(sk_s, m)$. Output $\sigma = \sigma' || sk_e$.
$\mathsf{BadCSE}(\mathcal{E}, \mathcal{S}).\mathsf{Verify}(pk, \sigma, m)$**:** Parse σ as $\sigma' || sk_e$. Run $\mathcal{S}.\mathsf{Verify}(pk_s, \sigma', m)$ and output the result.

$\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Encrypt}(pk, m)$**:** Output $c = \mathcal{E}.\mathsf{Encrypt}(pk_e, m)$.
$\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Decrypt}(sk, c)$**:** Run $\mathcal{E}.\mathsf{Decrypt}(sk_e, c)$. If this decryption is successful, output the decrypted message. Otherwise (if $\perp$ was returned), output sk_s.

From the security of the base schemes it is easy to see that the signature scheme given by the algorithms $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{KeyGen}$, $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Sign}$, $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Verify}$ is EUF-CMA secure, and the PKE scheme with algorithms $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{KeyGen}$, $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Encrypt}$, $\mathsf{BadCSE}(\mathcal{E},\mathcal{S}).\mathsf{Decrypt}$ is IND-CCA secure. However when key generation is shared a single signature reveals the PKE scheme's private key, and the decryption of a badly formed ciphertext reveals the private key of the signature scheme. Thus $\mathsf{BadCSE}(\mathcal{E},\mathcal{S})$ is totally insecure, even though its component schemes are secure.

4 A Generic Construction from IBE

We show how to build a combined signature and encryption scheme from an IBE scheme $\mathcal{I}$ with algorithms $\mathcal{I}.\mathsf{Setup}$, $\mathcal{I}.\mathsf{Extract}$, $\mathcal{I}.\mathsf{Encrypt}$, $\mathcal{I}.\mathsf{Decrypt}$. We make use of a one time strongly secure signature scheme $\mathcal{OT}$ with algorithms $\mathcal{OT}.\mathsf{KeyGen}$, $\mathcal{OT}.\mathsf{Sign}(sk, m)$, $\mathcal{OT}.\mathsf{Verify}(pk, \sigma, m)$. The construction is particularly simple: the signature scheme component is constructed through the Naor transform [6] and the PKE scheme component through the CHK transform [5]. Since in the Naor construction signatures are just private keys from the IBE scheme, and these private keys can be used to decrypt ciphertexts in the PKE scheme resulting from the CHK transform, we use a bit prefix in the identity space to provide domain separation between the signatures and private keys.

We assume $\mathcal{I}$ has message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and identity space $\{0,1\}^{n+1}$, and that $\mathcal{OT}$ has public key space $\{0,1\}^n$. Then the signature scheme component of $\mathsf{CSE}(\mathcal{I})$ has message space $\{0,1\}^n$ but can be extended to messages of arbitrary length through the use of a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$. The PKE component of $\mathsf{CSE}(\mathcal{I})$ has message space $\mathcal{M}$. The algorithms of $\mathsf{CSE}(\mathcal{I})$ are shown in Figure 1. In the full version [24] we show how the construction can be extended to support a tag-based encryption component.

Theorem 1 *Let $\mathcal{I}$ be a (t', q, ϵ)-OW-ID-CPA secure IBE scheme. Then the signature component of $\mathsf{CSE}(\mathcal{I})$ is (t, q_d, q_s, ϵ)-EUF-CMA secure in the presence of a decryption oracle provided that*

$$q_s + q_d \leq q \qquad \textit{and} \qquad t \leq t' - q_d(T_v + T_d) - T_d,$$

where T_v is the maximum time for a verification in $\mathcal{OT}$ and T_d is the maximum time for a decryption in $\mathcal{I}$.

Proof of Theorem 1. Suppose there exists a forger $\mathcal{F}$ that (t, q_d, q_s, ϵ) breaks the EUF-CMA security of the signature component of $\mathsf{CSE}(\mathcal{I})$ in the presence of a decryption oracle. We construct an algorithm $\mathcal{A}$ that interacts with the forger $\mathcal{F}$ to (t', q, ϵ)-OW-ID-CPA break the IBE scheme $\mathcal{I}$.

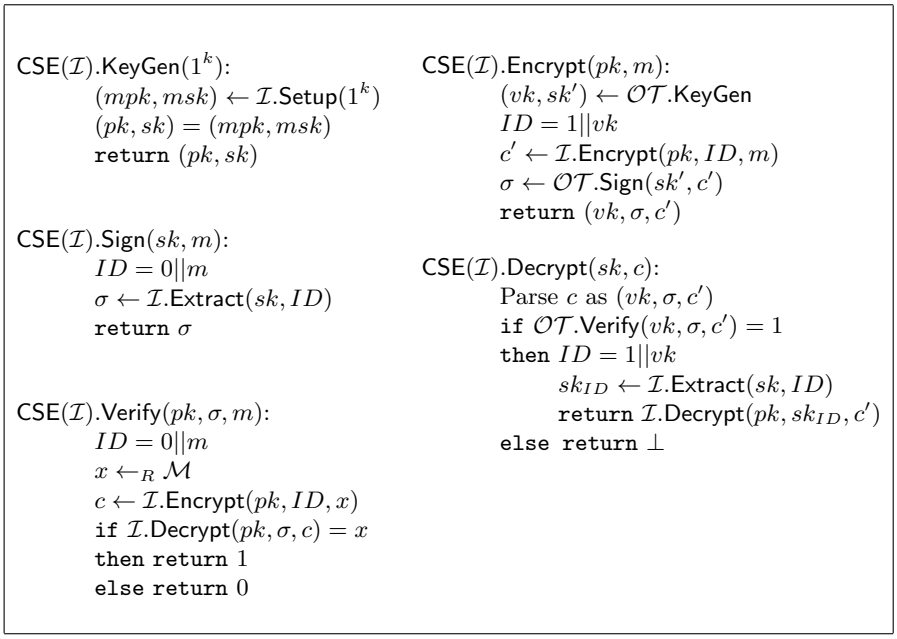

Fig. 1. Generic construction from IBE

Setup: $\mathcal{A}$ is given a master public key mpk which it gives to $\mathcal{F}$ as the public key.

Signing queries: In response to a request for a signature on message m, $\mathcal{A}$ queries its extraction oracle for the identity $ID = 0||m$ to obtain sk_{ID} which it returns to $\mathcal{F}$ as the signature.

Decryption queries: In response to a decryption query for a ciphertext $c = (vk, \sigma, c')$, $\mathcal{A}$ verifies that σ is a valid signature on c' with verification key vk. If it is not a valid signature, $\mathcal{A}$ returns $\perp$. If the signature is valid, $\mathcal{A}$ queries its extraction oracle for the identity $ID = 1||vk$ to obtain sk_{ID} which it uses to decrypt c', returning the output of the decryption operation as the result of the decryption query.

Forgery: Eventually $\mathcal{F}$ will return a forgery (σ^*, m^*) on a message m^* for which a signing query was not made. At this point $\mathcal{A}$ outputs $ID^* = 0||m^*$ as the target identity. This is a valid choice; since a signing query was not made for message m^* an extraction query was not made for $ID = 0||m^*$.

Challenge: $\mathcal{A}$ receives a ciphertext c^*, which is the encryption of a random message m for identity ID^*. If σ^* is a valid signature for message m^* then σ^* is a valid decryption key for identity ID^*. This allows $\mathcal{A}$ to decrypt c^* using $sk_{ID^*} = \sigma^*$ to retrieve the message m which it subsequently outputs.

$\mathcal{A}$ succeeds precisely when $\mathcal{F}$ succeeds, so if $\mathcal{F}$ outputs a valid forgery with probability ϵ in time t then algorithm $\mathcal{A}$ succeeds in time at most $t + q_d(T_v + T_d) + T_d$

with the same probability ϵ.

Theorem 2 *Let $\mathcal{I}$ be an (t_i, q_i, ϵ_i)-IND-sID-CPA secure IBE scheme and let $\mathcal{OT}$ be a (t_s, ϵ_s)-strongly unforgeable one time signature scheme. Then the encryption component of $\mathsf{CSE}(\mathcal{I})$ is (t, q_d, q_s, ϵ)-IND-CCA secure in the presence of a signing oracle provided that*

$$\epsilon > \frac{1}{2}\epsilon_s + \epsilon_i, \quad q_s + q_d < q_i, \quad \text{and} \quad t < t_i - T_{kg} - T_{sig} - q_d(T_v + T_d),$$

where T_{kg}, T_{sig} and T_v are the maximum times for key generation, signing and verifying respectively in $\mathcal{OT}$, and T_d is the maximum decryption time in $\mathcal{I}$.

Proof of Theorem 2. The proof follows closely that of Theorem 1 in [5]. Let $\mathcal{D}$ be an adversary against the IND-CCA security of the encryption component of $\mathsf{CSE}(\mathcal{I})$ in the presence of a signing oracle running in time at most t and making at most q_s signature queries and q_d decryption queries. We use $\mathcal{D}$ to build an IND-sID-CPA adversary $\mathcal{B}$ against $\mathcal{I}$ as follows.

Setup: $\mathcal{B}$ runs $\mathcal{OT}.\mathsf{KeyGen}$ to obtain a keypair (vk^*, sk^*) then submits $ID^* = 1||vk^*$ as the target identity. $\mathcal{B}$ is then given master public key mpk which it gives to $\mathcal{D}$ as the challenge public key.

Decryption queries: We partition the decryption queries into three possible cases and show how $\mathcal{B}$ responds to each case. Suppose the query is for ciphertext (vk, σ, c'), and let $\mathcal{OT}.\mathsf{Verify}(vk, \sigma, c') = validity$.

Case 1: $vk = vk^*$
If $validity = 0$ then $\mathcal{B}$ responds to the decryption query with $\perp$. If $validity = 1$ then a forgery has been made against $\mathcal{OT}$, call this event Forge. If Forge occurs, $\mathcal{B}$ aborts and outputs a random bit b'.

Case 2: $vk \neq vk^*$ and $validity = 0$
$\mathcal{B}$ responds to the decryption query with $\perp$.

Case 3: $vk \neq vk^*$ and $validity = 1$
$\mathcal{B}$ queries the extraction oracle for identity $ID = 1||vk$ to obtain sk_{ID}, then uses sk_{ID} to decrypt c', responding to the decryption query with the output of the decryption operation.

Signature queries: In response to a signature query for message m, $\mathcal{B}$ queries its extraction oracle for identity $ID = 0||m$ to obtain sk_{ID} which it returns as the signature.

Challenge: Eventually $\mathcal{D}$ will output a pair of messages m_0, m_1. $\mathcal{B}$ forwards these messages and receives a challenge ciphertext c^*. $\mathcal{B}$ calls $\mathcal{OT}.\mathsf{Sign}(sk^*, c^*)$ to obtain σ^* and sends $C = (vk^*, \sigma^*, c^*)$ to $\mathcal{D}$. $\mathcal{D}$ may make more signature and decryption queries under the restriction that it must not submit to the decryption oracle its challenge ciphertext C. $\mathcal{D}$ then submits a guess b' which $\mathcal{B}$ outputs as its guess.

$\mathcal{B}$ represents a legal strategy for attacking $\mathcal{I}$, in particular $\mathcal{B}$ never requests the private key corresponding to the target identity ID^*. Provided Forge does

not occur, $\mathcal{B}$ provides a perfect simulation for $\mathcal{D}$ so $\mathcal{B}$ succeeds with the same probability as $\mathcal{D}$. If Forge does occur then $\mathcal{B}$ outputs a random bit and succeeds with probability $\frac{1}{2}$. Letting $\Pr^{\mathcal{B}}_{\mathrm{IBE}}[\mathsf{Succ}]$ denote the probability of $\mathcal{B}$ outputting the correct bit in the IBE security game and $\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ}]$ denote the probability of $\mathcal{D}$ outputting the correct bit in the PKE security game, it can be seen that

$$\left|\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] - \frac{1}{2}\right| = \left|\Pr^{\mathcal{B}}_{\mathrm{IBE}}[\mathsf{Succ}] - \frac{1}{2}\right|.$$

Since $\mathcal{I}$ is an (t_i, q_i, ϵ_i)-IND-sID-CPA secure IBE scheme, $\left|\Pr^{\mathcal{B}}_{\mathrm{IBE}}[\mathsf{Succ}] - \frac{1}{2}\right| < \epsilon_i$. The event Forge represents a signature forgery against $\mathcal{OT}$, so $\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] < \epsilon_s$. It follows that

$$\begin{aligned}
\epsilon &= \left|\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ}] - \frac{1}{2}\right| \\
&\leq \left|\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ} \wedge \mathsf{Forge}] - \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}]\right| + \\
&\qquad\qquad \left|\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] - \frac{1}{2}\right| \\
&\leq \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] + \left|\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Succ} \wedge \overline{\mathsf{Forge}}] + \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] - \frac{1}{2}\right| \\
&= \frac{1}{2}\Pr^{\mathcal{D}}_{\mathrm{PKE}}[\mathsf{Forge}] + \left|\Pr^{\mathcal{B}}_{\mathrm{IBE}}[\mathsf{Succ}] - \frac{1}{2}\right| \\
&\leq \frac{1}{2}\epsilon_s + \epsilon_i.
\end{aligned}$$

The running time of $\mathcal{B}$ is at most $t + T_{kg} + q_d(T_v + T_d) + T_{sig}$, and it asks at most $q_s + q_d$ private key extraction queries, so the theorem holds.

IBE schemes meeting the standard model security requirements include those of Gentry [14] and Waters [28]. The latter results in a large public key (n+3 group elements), though this could be reduced in practice by generating most of the elements from a seed in a pseudo-random manner. We focus on the instantiation of our construction using Gentry's scheme. This scheme was originally presented in the setting of symmetric pairings. When we translate it to the asymmetric setting (see the full version for details) and apply our construction at the 128-bit security level using BN curves with sextic twists, we obtain a combined public key scheme in which the public key consists of two elements of $\mathbb{G}_1$ and two elements of $\mathbb{G}_2$, giving a public key size of 1536 bits. Ciphertexts encrypt elements of $\mathbb{G}_T$ and consist of an element of $\mathbb{G}_1$, two elements of $\mathbb{G}_T$, and a verification key and signature from $\mathcal{OT}$, so are 2304 bits plus the bit length of a verification key and signature in $\mathcal{OT}$. Signatures consist of an element of $\mathbb{Z}_p$ and an element of $\mathbb{G}_2$, so are 768 bits in size. Here we assume that descriptions of groups and pairings are domain parameters that are omitted from our key size calculations. The security of this scheme depends on an assumption closely related to the decisional q-augmented bilinear Diffie-Hellman exponent assumption.

This construction could be improved further using the Boneh-Katz [5] alternative to the CHK transform. We omit the details in favour of our next scheme.

5 A More Efficient Construction

The following scheme is based on the signature scheme by Boneh and Boyen [4] and a KEM obtained by applying the techniques by Boyen, Mei and Waters [7] to the second IBE scheme by Boneh and Boyen in [3]. The schemes make use of a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where the groups are of order p, and the KEM furthermore makes use of an always second-preimage resistant (aSec-secure) hash function $\mathsf{H} : \mathbb{G}_1 \to \{0,1\}^{n-1}$ where $2^n < p$. To obtain a full encryption scheme, the KEM is combined with a DEM, and we assume for simplicity that the key space of the DEM is $\mathcal{K} = \mathbb{G}_T$. Where a binary string is treated as a member of $\mathbb{Z}_p$ it is implicitly converted in the natural manner. The signature scheme supports messages in $\{0,1\}^{n-1}$, but can be extended to support message in $\{0,1\}^*$ by using a collision resistant hash function, while the encryption scheme supports messages of arbitrary length due to the use of a DEM. Note that to minimize the public key size and ciphertext overhead in the scheme, the elements of the public key are placed in the group $\mathbb{G}_1$. However, this implies that signatures contain an element of the group $\mathbb{G}_2$, having larger bit representations of elements.

$\mathsf{KeyGen}(1^k)$: Choose random generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and random integers $x, y \in \mathbb{Z}_p^*$, and compute $X = g_1^x$ and $Y = g_1^y$. The public key is (g_1, g_2, X, Y) and the private key is (x, y).

$\mathsf{Sign}(sk, m)$: To sign a message $m \in \{0,1\}^{n-1}$ first prepend a zero to m to give $m' = 0||m \in \{0,1\}^n$. Choose random $r \in \mathbb{Z}_p$. If $x + ry + m' \equiv 0 \bmod p$ then select another $r \in \mathbb{Z}_p$. Compute $\sigma = g_2^{\frac{1}{x+m'+yr}} \in \mathbb{G}_2$. The signature is $(\sigma, r) \in \mathbb{G}_2 \times \mathbb{Z}_p$.

$\mathsf{Verify}(pk, \sigma, m)$: If $e(X \cdot g_1^{m'} \cdot Y^r, \sigma) = e(g_1, g_2)$, where $m' = 0||m$, then return 1, otherwise return 0.

$\mathsf{Encrypt}(pk, m)$: To encrypt a message $m \in \{0,1\}^*$, choose random $s \in \mathbb{Z}_p^*$ and compute $c_1 = Y^s$ and $h = \mathsf{H}(c_1)$. Prepend a 1 to h to give $h' = 1||h \in \{0,1\}^n$, and compute $c_2 = X^s \cdot g_1^{s \cdot h'}$. Lastly, compute the key $K = e(g_1, g_2)^s \in \mathbb{G}_T$ and encrypt the message m using the DEM i.e. $c_3 = \mathsf{DEnc}(K, m)$. The ciphertext is $c = (c_1, c_2, c_3)$.

$\mathsf{Decrypt}(sk, c)$: To decrypt a ciphertext $c = (c_1, c_2, c_3)$, first compute $h = \mathsf{H}(c_1)$ and prepend a 1 to h to get $h' = 1||h$. If $c_1^{(x+h')/y} \neq c_2$, output $\perp$. Otherwise, compute the key $K = e(c_1, g_2^{1/y}) \in \mathbb{G}_T$, and output the message $m = \mathsf{DDec}(K, c_3)$.

We note that the computational cost of encryption and signature verification can be reduced by adding the redundant element $v = e(g_1, g_2)$ to the public key, but that this will significantly increase the public key size.

Theorem 3 *Suppose the (t', q, ϵ')-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$. Then the above combined public key scheme is (t, q_d, q_s, ϵ)-EUF-CMA secure in the presence of a decryption oracle given that*

$$q_s \leq q, \quad \epsilon \geq 2\epsilon' + q_s/p \approx 2\epsilon' \quad \text{and} \quad t \leq t' - \Theta(q_d T_p + (q_d + q^2)T_e),$$

where T_p is the maximum time for evaluating a pairing and T_e is the maximum time for computing an exponentiation in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{Z}_p$.

Theorem 4 *Suppose that the hash function H is (t_h, ϵ_h)-aSec secure, that the $(t_{dhi}, q_{dhi}, \epsilon_{dhi})$-DBDHI assumption holds in the groups $\mathbb{G}_1, \mathbb{G}_2$, and that the DEM is $(t_{dem}, q_{dem}, \epsilon_{dem})$-IND-CCA secure. Then the combined public key scheme above is $(t, q_d, q_s, \epsilon,)$-IND-CCA secure in the presence of a signing oracle given that*

$$q_s \leq q_{dhi}, \quad q_d \leq q_{dem}, \quad \epsilon \geq \epsilon_h + \epsilon_{dhi} + \epsilon_{dem} + q_s/p, \quad \text{and}$$
$$t \leq t_{min} - \Theta(q_d T_p + (q_{dhi} + q_d)T_e),$$

where $t_{min} = \min(t_h, t_{dhi}, t_{dem})$, T_p is the maximum time for evaluating a pairing, and T_e is the maximum time for computing an exponentiation in $\mathbb{G}_1, \mathbb{G}_2$.

The proofs of Theorems 3 and 4 can be found in the full version [24].

The above scheme provides public keys consisting of three group elements of $\mathbb{G}_1$ and one group element of $\mathbb{G}_2$. If the scheme is instantiated using BN curves with sextic twists mentioned above, this translates into a public key size of 1280 bits for a 128 bit security level. Furthermore, assuming that the DEM is redundancy-free (which can be achieved if the DEM is a strong pseudorandom permutation [25]), the total ciphertext overhead is just two group elements of $\mathbb{G}_1$ which translates into 512 bits. Signatures consist of a single group element of $\mathbb{G}_2$ and an element of $\mathbb{Z}_p$, and will be 768 bits. Again, we assume that descriptions of groups and pairings are ignored in these calculations.

In the full version, we show how the construction can be extended to support tag-based encryption. This property is required to allow us to use the scheme to instantiate our combined signcryption, signature and encryption scheme (see the full version for details).

6 Comparison of Schemes

In this section, we provide a comparison of the schemes arising from our IBE-based construction, our more efficient construction in Section 5 and the Cartesian product construction. In our comparison we will limit ourselves to other discrete-log/pairing-based schemes since provably secure (standard model) lattice-based schemes with short public keys are still unavailable and factoring-based schemes do not scale very well (for 128-bit security, the modulus would need to be > 3000 bits which is not competitive). We will include group generators in public key size calculations as the required number depends on the scheme, but we allow

sharing of generators between signature and encryption component in Cartesian product instantiations to improve these constructions. Note that it is possible to reduce the private key of any scheme to a single short random seed by making the following simple modification to the scheme: to generate a public/private keypair, pick a random seed, generate the randomness required by the key generation algorithm by applying a pseudorandom generator to the seed, and generate the public/private keypair using this randomness, but store only the seed as the private key. Whenever the original private key is needed, re-compute this by applying the pseudorandom generator to the seed and re-run the key generation algorithm with the resulting randomness. This observation essentially makes the difference in private key sizes irrelevant, and we will not include this aspect in our comparison. We consider several instantiations of the Cartesian product construction with standard model secure encryption and signature schemes and give the results in Figure 2.

We will focus on Cartesian product instantiations using the scheme by Boneh and Boyen [4] as a signature component. This scheme is among the most efficient signature schemes and additionally has a short public key. To reduce the public key size even further, we can remove the redundant element $v = e(g_1, g_2)$ and place as many elements as possible in the group $\mathbb{G}_1$ of the pairing. The latter implies that signatures will be elements of $\mathbb{G}_2 \times \mathbb{Z}_p$ which results in an increase in signature size. However, since the Cartesian product constructions should compete with the combined public key schemes in terms of public key size, this tradeoff is desirable. While other signature schemes could be considered, we were not able to find a scheme providing shorter public keys without a significant disadvantage elsewhere. For instance, hash-based signature schemes give extremely short public keys (the hash function description plus the root digest), but result in signatures with length logarithmic in the number of messages to be signed. The signature scheme by Hofheinz and Kiltz [17] has shorter signatures than the Boneh-Boyen scheme and a public key consisting of a few group elements plus a hash key, but here the hash key will be long to achieve provable programmability.

For the encryption component, a relevant option is a DEM combined with the KEM obtained by applying the techniques by Boyen, Mei and Waters [7] to the second IBE scheme of Boneh and Boyen in [3], which also forms the basis of our concrete scheme. Combined with the Boneh-Boyen signature scheme, and assuming the group generators in the two schemes are shared, this yields a very efficient instantiation of the Cartesian product construction in which public keys consist of five group elements of $\mathbb{G}_1$, one group element of $\mathbb{G}_2$ (and a key defining a target collision resistant hash function). This is larger by two elements of $\mathbb{G}_1$ than the public key in our concrete construction from Section 5, which translates to a difference of 512 bits. Note that signature size, ciphertext overhead and computation costs are the same for the Cartesian product scheme and our construction.

Another encryption scheme to consider is that of Kurosawa and Desmedt [22]. Instantiating the Cartesian product construction with this scheme and the

Signature Scheme	PKE Scheme	Public Key Size	Signature Size	Ciphertext Overhead
BB [4]	BB [3] + BMW [7]	1792	768	512
BB [4]	KD [22]	2048	768	640
BB [4]	Kiltz [19]	1792	768	512
CSE(Gentry)		1536	768	$1280 + \|vk_{\mathcal{OT}}\| + \|\sigma_{\mathcal{OT}}\|$
Scheme from Sec. 5		1280	768	512

Fig. 2. Comparison of schemes at the 128-bit security level.

Boneh-Boyen signature scheme yields a scheme with a public key consisting of six elements of $\mathbb{G}_1$, one element of $\mathbb{G}_2$ (and a key defining a target collision resistant hash), assuming that the Kurosawa-Desmedt scheme is implemented in $\mathbb{G}_1$. Hence, the public key will be larger by three group elements of $\mathbb{G}_1$ compared to our concrete construction, which equates to a difference of 768 bits at the 128-bit security level. Signature size and signing and verification costs will be the same as in our construction, whereas the ciphertext overhead will be slightly larger (an extra 128 bits) due to the requirement that the symmetric encryption scheme used in the Kurosawa-Desmedt scheme is authenticated. However, decryption costs will be lower since no pairing computations are required.

Lastly, the encryption scheme of Kiltz [19] might be considered. Again, combining this with the Boneh-Boyen signature scheme, and assuming group generators are shared, will yield a Cartesian product scheme with public keys consisting of five elements of $\mathbb{G}_1$ and one element of $\mathbb{G}_2$. This is two group elements of $\mathbb{G}_1$ larger than the public key of our concrete construction, which equates to an increase of 512 bits at the 128-bit security level. Signature size and ciphertext overhead will be the same while decryption in the Cartesian product scheme will be more efficient, since no pairing computations are required.

In summary, our concrete construction of a combined public key scheme admits shorter public keys than any instantiation of the Cartesian product construction of Section 3.1 with known standard model secure encryption and signature schemes, and furthermore enjoys compact ciphertexts and signatures.

7 Conclusions and Future Research

We have revisited the topic of joint security for combined public key schemes, focussing on the construction of schemes in the standard model, an issue not fully addressed in prior work. We gave a general construction for combined public key schemes from weakly secure IBE, as well as a more efficient concrete construction based on pairings. Using BN curves, these can be efficiently instantiated at high security levels and have performance that is competitive with the best schemes arising from the Cartesian product construction. Our results fill the gap left open in the original work of Haber and Pinkas [15], of constructing standard-model-secure combined public key schemes in which the signature and

encryption components share an identical keypair. An interesting open problem is to construct efficient combined public key schemes in the standard model not using pairings. For example, is it possible to obtain joint security in the discrete log or in the RSA setting, in the standard model?

Our work points the way to an interesting new research area in cryptography, which closely relates to and generalises the topic of cryptographic agility [1]. The general question can be posed as follows: under what conditions is it safe to use the same key (or key pair) across multiple instantiations of the *same* or *different* cryptographic primitives?

References

1. Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic agility and its relation to circular encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 403–422. Springer, Heidelberg (2010)
2. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S.E. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006)
3. Boneh, D., Boyen, X.: Efficient selective identity-based encryption without random oracles. Journal of Cryptology To appear, available from `http://www.springerlink.com/content/n63632331k4q4h11/`
4. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptology 21(2), 149–177 (2008)
5. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)
7. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM Conference on Computer and Communications Security. pp. 320–329. ACM (2005)
8. Coron, J.S., Joye, M., Naccache, D., Paillier, P.: Universal padding schemes for RSA. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 226–241. Springer, Heidelberg (2002)
9. Dodis, Y., Freedman, M.J., Jarecki, S., Walfish, S.: Optimal signcryption from any trapdoor permutation. Cryptology ePrint Archive, Report 2004/020 (2004), `http://eprint.iacr.org/`
10. Dodis, Y., Freedman, M.J., Jarecki, S., Walfish, S.: Versatile padding schemes for joint signature and encryption. In: ACM Conference on Computer and Communications Security. pp. 344–353. ACM (2004)
11. EMV Specifications, Version 4.2, Books 1–4 (June 2008), `http://www.emvco.com/`
12. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptology 23(2), 224–280 (2010)
13. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)
14. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
15. Haber, S., Pinkas, B.: Securely combining public-key cryptosystems. In: ACM Conference on Computer and Communications Security. pp. 215–224 (2001)

16. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)
17. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
18. Kelsey, J., Schneier, B., Wagner, D.: Protocol interactions and the chosen protocol attack. In: Christianson, B., Crispo, B., Lomas, T.M.A., Roe, M. (eds.) Security Protocols Workshop. LNCS, vol. 1361, pp. 91–104. Springer, Heidelberg (1997)
19. Kiltz, E.: Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
20. Klíma, V., Rosa, T.: Further results and considerations on side channel attacks on RSA. In: Jr., B.S.K., Çetin Kaya Koç, Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 244–259. Springer, Heidelberg (2003)
21. Komano, Y., Ohta, K.: Efficient universal padding techniques for multiplicative trapdoor one-way permutation. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 366–382. Springer, Heidelberg (2003)
22. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
23. Matsuda, T., Matsuura, K., Schuldt, J.C.N.: Efficient constructions of signcryption schemes and signcryption composability. In: Roy, B.K., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 321–342. Springer, Heidelberg (2009)
24. Paterson, K.G., Schuldt, J.C., Stam, M., Thomson, S.: On the joint security of encryption and signature, revisited. Cryptology ePrint Archive, Report 2011/??? (2011)
25. Phan, D.H., Pointcheval, D.: About the security of ciphers (semantic security and pseudo-random permutations). In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 182–197. Springer, Heidelberg (2005)
26. Rubin, K., Silverberg, A.: Compression in finite fields and torus-based cryptography. SIAM J. Comput. 37(5), 1401–1428 (2008)
27. Vasco, M.I.G., Hess, F., Steinwandt, R.: Combined (identity-based) public key schemes. Cryptology ePrint Archive, Report 2008/466 (2008), `http://eprint.iacr.org/`
28. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)