Combined (identity-based) public key schemes

María Isabel González Vasco[1], Florian Hess[2], and Rainer Steinwandt[3]

[1] Departamento de Matemática Aplicada, Universidad Rey Juan Carlos,
c/ Tulipán, s/n, 28933 Madrid, Spain;
mariaisabel.vasco@urjc.es
[2] Technische Universität Berlin,
Fakultät II – Institut für Mathematik Sekr. MA 8-1,
Straße des 17. Juni 136, 10623 Berlin, Germany
hess@math.tu-berlin.de
[3] Department of Mathematical Sciences, Florida Atlantic University,
777 Glades Road, Boca Raton, FL 33431, USA
rsteinwa@fau.edu

Abstract. Consider a scenario in which parties use a public key encryption scheme and a signature scheme with a single public key/private key pair—so the private key *sk* is used for both signing and decrypting. Such a simultaneous use of a key is in general considered poor cryptographic practice, but from an efficiency point of view looks attractive.

We offer security notions to analyze such violations of key separation. For both the identity- and the non-identity-based setting, we show that—although being insecure in general—for schemes of interest the resulting *combined (identity-based) public key scheme* can offer strong security guarantees.

Keywords: Combined public key scheme, identity-based cryptography, key separation.

1 Introduction

Schemes for public key encryption and digital signatures are among the most prominent cryptographic tools, and often both of these primitives are used within the same protocol. Although a digital signature scheme and a public key encryption scheme aim at different goals, the underlying algorithms may have much in common, and the question for synergies when implementing both primitives arises. From an efficiency point of view it is desirable that resources can be shared among the two schemes: if both schemes rely on a similar computational assumption, it may well happen that the key generation procedure is verbatim identical. For the sake of efficiency, here it is tempting to use a single public key/private key pair for both the signature and the encryption scheme. However, this clearly violates the principle of *key separation* and is commonly considered poor cryptographic practice.

Related work. If the essential application of an encryption and a signature scheme in a protocol consists of signing messages with a sender's private key followed by encrypting the signed messages under a recipient's public key, then signcryption [22] can be an attractive *alternative* to separate encryption and signature mechanisms. For a signcryption scheme, the decryption and signing keys are typically from different parties, but insider security against multi-user signcryption [1, 12] is analogous to our security goal in the sense that both indistinguishability of ciphertexts and existential unforgeability must be achieved. While a signcryption scheme is designed to satisfy this type of requirement, here we glue a given signature scheme to a given encryption scheme by enforcing the use of a single key pair for both schemes, and we try to understand the security implications of this glueing. As detailed in [1], a signcryption scheme in particular induces a signature and an encryption scheme. With regard to key lengths, however, these induced schemes appear inferior to dedicated encryption or signature mechanisms, as essentially two signcryption keys are used to form one key for the induced signature or encryption scheme. For a scenario where we want the flexibility of separate encryption and signature mechanisms, the use of a signcryption scheme appears less attractive than a "secure key reuse" as described below.

One plausible approach to explore the double use of keys is to invoke the universal composition (UC) framework [8]. More specifically, one could try to interpret the problem we address in this paper as a *composition with joint state* [9]. We are not aware that a key "double use" as we discuss it has been explored in the UC framework so far. This also applies to the identity based setting: *universally composable identity-based encryption* as considered by Nishimaki et al. in [19] does not guarantee that a single key for signature and encryption can be used securely: According to [19, Theorem 1] non-adaptively realizing the functionality $\mathcal{F}_{\mathrm{IBE}}$ from [19] is equivalent to IND-ID-CCA-security[1].

Known results. For the concrete scenario considered in this paper, Haber and Pinkas [16] show that the simultaneous use of related keys in a signature scheme and a public key encryption scheme is, for several examples, secure in a strong sense. More specifically, they consider an adversary against a signature scheme which has (unrestricted) access to a decryption oracle of an encryption scheme using a related secret key, and prove that for several signature schemes such adversaries are not more damaging than "standard" ones. Analogously, for some encryption schemes, they prove that an attacker who is granted (unrestricted) access to a signing oracle of a signature scheme using a related secret key will not endanger the security of the encryption scheme. In subsequent work, a main focus was on the design of *universal padding schemes* that can be used for both signing and encryption without the need of separate keys. Coron et al. showed that PSS enables such a secure composition of a signature and encryption scheme with a single key pair [11]. More recently, Komano and Ohta propose combined constructions building on OAEP+ and REACT. Instead of the *partial-domain one-wayness* requirement of Coron et al., [18] imposes a one-wayness require-

[1] In this paper, we use CCA synonymously with CCA2.

ment only. Further refinements of universal paddings are explored by Chevallier-Mames et al. in [10].

Our Contribution. Section 2 follows Haber and Pinkas [16] in the sense that we try to combine existing schemes that have not been designed for usage with a common private key. We analyze the security of such *combined schemes* using dedicated security notions building on the ones coined by Komano and Ohta in [18]. After showing how the simultaneous use of a private key can be fatal, we give a *combined scheme* with a security proof: already [16, Section 4.5] notes that the ElGamal signature scheme in the modification of Pointcheval and Stern [21] would remain secure if an adversary had access to a decryption oracle for an encryption scheme using the same private key. We combine this signature scheme with an ElGamal encryption scheme under a Fujisaki-Okamoto conversion and prove the resulting scheme to be secure in a strong sense: in the random oracle model, the combined scheme achieves both existential unforgeability and indistinguishability of encryptions.

Section 3 explores the use of a unique private key in an identity-based setting: an identity-based public key encryption scheme IBE and an identity-based signature scheme IBS share a setup and key extraction algorithm and each user has one secret key only which is used for both signing and decrypting. We prove that such a simultaneous use can be possible without jeopardizing the security of the involved schemes. Namely, for an identity based signature scheme by Hess [17] and an identity based encryption scheme of Boneh and Franklin [7] we prove security in the sense of a natural generalization of standard security notions in identity-based cryptography.

2 Combined public key schemes

2.1 Preliminaries and definitions

Adapting the terminology from [16], we define a *combined public key scheme* as a combination of a public key encryption scheme and a signature scheme that have the key generation in common:

Definition 1 (combined public key scheme).
A combined public key scheme *is a tuple* $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ *of polynomial time algorithms:*

- $\mathcal{K}$ *is a probabilistic* key generation algorithm *that upon input the security parameter* 1^k *outputs a public key/secret key pair* (pk, sk).
- $\mathcal{E}$ *is a probabilistic* encryption algorithm *that upon input a message* m *and a public key* pk *computes a ciphertext* $c \leftarrow \mathcal{E}_{pk}(m)$.
- $\mathcal{D}$ *is a deterministic* decryption algorithm *that upon input a candidate ciphertext* c *and a secret key* sk *outputs a plaintext* $m \leftarrow \mathcal{D}_{sk}(c)$ *or an error symbol* $\perp$.
- $\mathcal{S}$ *is a probabilistic* signing algorithm *that upon input a message* m *and a secret key* sk *outputs a signature* $\sigma \leftarrow \mathcal{S}_{sk}(m)$.

– *$\mathcal{V}$ is a deterministic* verification algorithm *that upon input a public key pk, a message m and a candidate signature σ outputs* true *or* false.

For a pair (pk, sk) generated by $\mathcal{K}$ we require that with overwhelming probability the obvious correctness condition holds: For all messages m we have $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)) = m$ and $\mathcal{V}_{pk}(m, \mathcal{S}_{sk}(m)) = \mathsf{true}$.

To model the security of a combined public key scheme we adapt the notions of EUF-CCA2&ACMA and IND-CCA2&ACMA security introduced in [18]. For brevity, we deviate from [18] and simply write EUF-C[CM]A respectively IND-C[CM]A when considering "adversaries with an additional oracle." Essentially, the notion IND-C[CM]A formalizes the situation in which an IND-CCA adversary has, in addition to the usual tools, access to a signing oracle, and, analogously, an EUF-C[CM]A adversary is an EUF-CMA adversary having access to a decryption oracle, too.:

Definition 2 (IND-C[CM]A).
Let $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ be a combined public key scheme, and let $\mathcal{A}$ be a probabilistic polynomial time adversary. Consider the following attack scenario:

1. *Compute a key pair $(pk, sk) \leftarrow \mathcal{K}(1^k)$, and hand pk as input to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ is given unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ to run $\mathcal{S}_{sk}(\cdot)$ and unrestricted access to a decryption oracle $\mathcal{O}_{\mathcal{D}}$ to run $\mathcal{D}_{sk}(\cdot)$. At the end of this stage, $\mathcal{A}$ outputs two plaintexts $m_0 \neq m_1$ of equal length.*
3. *A value $b \in_R \{0, 1\}$ is chosen uniformly at random, and $\mathcal{A}$ learns a target ciphertext $c \leftarrow \mathcal{E}_{pk}(m_b)$.*
4. *The algorithm $\mathcal{A}$ is again given unrestricted access to the signing oracle $\mathcal{O}_{\mathcal{S}}$, and the only restriction in querying $\mathcal{O}_{\mathcal{D}}$ is that target ciphertext c must not be queried. At the end of this stage $\mathcal{A}$ outputs a guess b' for b.*

The advantage $\mathrm{Adv}_{\mathcal{A}} = \mathrm{Adv}_{\mathcal{A}}(k)$ *of $\mathcal{A}$ is defined as $|2 \cdot P[b = b'] - 1|$, and we call $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$* secure in the sense of IND-C[CM]A *if $\mathrm{Adv}_{\mathcal{A}}$ is negligible for all probabilistic polynomial time adversaries $\mathcal{A}$.*

Definition 3 (EUF-C[CM]A).
Let $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ be a combined public key scheme, and let $\mathcal{A}$ be a probabilistic polynomial time adversary. Consider the following attack scenario:

1. *Compute a key pair $(pk, sk) \leftarrow \mathcal{K}(1^k)$, and hand pk as input to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ is given unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ to run $\mathcal{S}_{sk}(\cdot)$ and unrestricted access to a decryption oracle $\mathcal{O}_{\mathcal{D}}$ to run $\mathcal{D}_{sk}(\cdot)$. At the end of this stage, $\mathcal{A}$ outputs a message m and a signature σ such that m has not been submitted to the signing oracle $\mathcal{O}_{\mathcal{S}}$.*

The success probability $\mathrm{Succ}_{\mathcal{A}} = \mathrm{Succ}_{\mathcal{A}}(k)$ *of $\mathcal{A}$ is defined as $P[\mathcal{V}_{pk}(m, \sigma) = \mathsf{true}]$, and we call $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$* secure in the sense of EUF-C[CM]A *if $\mathrm{Succ}_{\mathcal{A}}$ is negligible for all probabilistic polynomial time adversaries $\mathcal{A}$.*

2.2 Combining secure schemes is not sufficient

As already indicated with the definitions in the previous section, our main focus is on the combination of IND-CCA-secure encryption and EUF-CMA-secure signature schemes. Before looking at this in more detail, we note that for passive adversaries the situation is somewhat trivial:

*Remark 1 (*IND-CPA+EUF-NMA*).* Suppose we have a signature scheme that is secure against no message attacks (EUF-NMA) and a public key encryption scheme that is secure against chosen plaintext attacks (IND-CPA). If these schemes have an identical key generation algorithm $\mathcal{K}$, then the resulting combined scheme certainly is secure against adversaries without access to a decryption or a signing oracle: the simultaneous use of the two schemes has no effect on the tools available to an adversary.

For adversaries with access to stronger tools, different situations can arise:

*Example 1 (*IND-CPA+EUF-CMA*).* We build on prominent encryption and signature constructions of Bellare and Rogaway [5, 4].

Signature Scheme: Denote by $f : \{0,1\}^k \longrightarrow \{0,1\}^k$ a trapdoor one-way permutation and by $H : \{0,1\}^k \longrightarrow \{0,1\}^k$ a random oracle.
- $\mathcal{K}$: outputs, as public key pk, enough information to efficiently evaluate f, and the secret key sk, which consists of trapdoor information to efficiently evaluate f^{-1};
- $\mathcal{S}$: to sign message $m \in \{0,1\}^k$ with secret key sk, $\mathcal{S}_{sk}(m)$ outputs the signature
$$(\sigma_1, \sigma_2) := (f^{-1}(H(m)), H(f^{-1}(m)));$$
- $\mathcal{V}$: to verify a signature (σ_1', σ_2') on message m for the public key pk, $\mathcal{V}_{pk}(m, (\sigma_1', \sigma_2'))$ returns true if and only if $f(\sigma_1') = H(m)$.[2]

Encryption Scheme: Let f and H be as before.
- $\mathcal{K}$: as above;
- $\mathcal{E}$: we set $\mathcal{E}_{pk}(m) := (f(r), H(r) \oplus m)$ with a uniformly at random chosen $r \in_R \{0,1\}^k$;
- $\mathcal{D}$: to decrypt $(f(r), H(r) \oplus m)$, we compute $H(f^{-1}(f(r))) \oplus (H(r) \oplus m) = m$.

We can easily show these two schemes to be secure in the sense of EUF-CMA and IND-CPA, respectively:

Lemma 1. *The signature scheme described in Example 1 is secure in the sense of* EUF-CMA *in the random oracle model.*

[2] Obviously, this signature scheme is *malleable* in the sense that from a given signature (σ_1, σ_2) for a message m, further signatures (σ_1, σ_2') for m can be derived.

Proof. We structure the proof along a sequence of games, at which the adversary $\mathcal{A}$, a probabilistic polynomial time adversary with unrestricted access to the random oracle H, interacts with a simulator/challenger. At this, we start in G_0 with the real attack simulation game, i. e., the adversary and the simulator interact as described in the standard EUF-CMA setting. In the sequel, event S_i denotes the event in which $\mathcal{A}$ wins the game against the challenger at Game G_i.

Game G_0. This is the real attack game. Thus, the challenger

1. Computes a key pair $(f, f^{-1}) \leftarrow \mathcal{K}(1^k)$, and hands f as input to $\mathcal{A}$.
2. $\mathcal{A}$ is given unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ to run $\mathcal{S}_{sk}(\cdot)$. At the end of this stage, $\mathcal{A}$ outputs a message m and a signature pair (σ_1, σ_2) such that m has not been submitted to the signing oracle $\mathcal{O}_{\mathcal{S}}$.

The probability of success the adversary has in this game, $P[S_0]$, is

$$P[\mathcal{V}_f(m, (\sigma_1, \sigma_2)) = \mathsf{true}].$$

Game G_1. We formalize at this the random oracle simulation of H: H-queries are answered by a simulator which keeps record of hash queries on a so-called H-list, i. e., for every query $H(\mathsf{query})$ the simulator outputs a value random selected uniformly at random from X and adds the triple $(\mathsf{num}, \mathsf{query}, \mathsf{random})$ to his H-list. Subsequently, for a hash query $H(x)$ such that a record (n, x, r) appears in the H-list (where n is any natural number), the answer of the simulator will be r.

Now, the random oracle assumption states that $P[S_0] = P[S_1]$.

Game G_2. We modify the simulation of the random oracle H: on input any value query, now the simulation of the random oracle outputs not only the corresponding value r as above, but also the output value of an H-query as in Game G_1 on input $f^{-1}(\mathsf{query})$—namely, now an H-query simulation involves two H-queries from the previous game. It is clear that this change does not augment the probability of success of our adversary, for, if he knows already $f^{-1}(m)$, this modified H-simulation does not provide him with anything new and, if he does not know $f^{-1}(m)$, $H(f^{-1}(m))$ looks like a randomly selected element from X to him. As a result, we still have $P[S_0] = P[S_1] = P[S_2]$.

From this point on, it is easy to see that our adversary is actually facing an EUF-CMA game against Bellare and Rogaway's FDH signature scheme, and, as a result, it follows that $P[S_2]$ is negligible in the security parameter. However, we reproduce here the arguments needed to complete the proof.

Game G_3. Let us modify the previous game in the sense that, in order to win the game, we require that the adversary not only produces a valid triple $(m^*, \sigma_1^*, \sigma_2^*)$, but moreover guesses correctly the position num m* at which m^* occurs in the H-list. (Note that we may assume m^* has indeed been queried to the random oracle, as the triple $(m^*, \sigma_1^*, \sigma_2^*)$ must at some point pass the verification process).

Now

$$P[S_3] \geq P[S_2] \cdot \frac{1}{p(k)},$$

where $p(k)$ is a polynomial upper bound on the number of queries of $\mathcal{A}$ to $\mathcal{O}_{\mathcal{S}}$ and H.

Game G_4. We further modify the simulation of the H-queries, in the following sense: for the query number $\mathsf{num\ m*}$, set $\mathsf{random} := y$. For any other query, define $\mathsf{random} := f(x)$ where x is selected uniformly at random from $\{0,1\}^k$. Exactly as before, the records $(\mathsf{num}, \mathsf{query}, \mathsf{random})$ are written down on the H-list.

Note that this simulation of H is indistinguishable from the one in Game G_3, as f is a permutation.

Game G_5. By now, the simulation of the signing oracle can be done without using the secret key, for all preimages of elements ever queried to H are known by the simulator (note that m^* is never queried to the signing oracle, as it is part of the forgery). We thus modify our simulation in that sense, and obviously $P[S_5] = P[S_4]$.

Moreover, it is clear that $(m, \sigma_1^*, \sigma_2^*)$ being a successful forgery actually implies $\mathcal{A}$ has been able to compute a preimage of y, and as a result, $P[S_5]$ is negligible, which concludes our proof. □

Lemma 2. *The encryption scheme described in Example 1 is secure in the sense of* IND-CPA *in the random oracle model.*

Proof. Actually, this scheme was first proposed and proven secure in [4]. We sketch a proof here for completeness: We formalize the proof via a sequence of games or experiments, in which the adversary interacts with a simulator. As usual, we will denote by S_i the event that the adversary wins in Game G_i.
Game G_0. Real attack game. Thus, the challenger

1. Computes a key pair $(pk, sk) \leftarrow \mathcal{K}(1^k)$, and hands pk as input to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ is given unrestricted access to a decryption oracle $\mathcal{O}_{\mathcal{D}}$ to run $\mathcal{D}_{sk}(\cdot)$. At the end of this stage, $\mathcal{A}$ outputs two plaintexts $m_0 \neq m_1$ of equal length.
3. A value $b \leftarrow \{0,1\}$ is chosen uniformly at random, and $\mathcal{A}$ learns a target ciphertext $c \leftarrow \mathcal{E}_{pk}(m_b)$.
4. The algorithm $\mathcal{A}$ is again given unrestricted access to the signing oracle $\mathcal{O}_{\mathcal{S}}$, and the only restriction in querying $\mathcal{O}_{\mathcal{D}}$ is that target ciphertext c must not be queried. At the end of this stage, $\mathcal{A}$ outputs a guess b' for b.

Now, as standard, $P[S_0] = |2 \cdot P[b = b'] = 1|$.

Game G_1. Usual random oracle simulation: H-queries are answered by a simulator which keeps record of hash queries on a so-called H-list, i. e., for every query $H(\mathsf{query})$ it outputs a value random selected uniformly at random from X and adds the triple $(\mathsf{num}, \mathsf{query}, \mathsf{random})$ to his H-list.

Subsequently, for a hash query $H(x)$ such that a record (n, x, r) appears in the H-list (where n is any natural number), the answer of the simulator will be r.

Again, the random oracle assumption states that $P[S_0] = P[S_1]$.

Game G_2. We modify the construction of the challenge ciphertext $c^* = (c_1^*, c_2^*)$ as follows:

– select $r \in \{0,1\}^k$ uniformly at random, define $c_1^* := f(r)$,
– select uniformly at random a bitstring mask, define $c_2^* := \mathsf{mask} \oplus m_b$.

Note that the adversary will only notice the difference between G_0 and G_1 if he is able to compute an f-preimage for c_1^*, i. e., with negligible probability. On the other hand, c_2^* is a one-time pad encryption of m_b, and, as a result, the adversary has probability $\frac{1}{2}$ of guessing correctly whether $b = 1$ or $b = 0$. $\square$

However, when combining the above schemes in the obvious way, the signing oracle is a powerful tool to violate the security of the encryption scheme. Indeed, given a challenge ciphertext $(f(r), H(r) \oplus m_b)$, the adversary can query a signature on $f(r)$ to learn the masking value $H(r)$ and thus recover the corresponding plaintext m_b.

The encryption scheme involved in the construction above was "only" secure in the sense of IND-CPA, and one might be tempted to think that security problems would not arise if both the encryption and signature schemes are more robust. However, we conclude this section with an example showing that combining an IND-CCA secure public key encryption scheme and an EUF-CMA secure signature scheme is in general not sufficient to obtain a combined public key scheme that is secure in the sense of IND-C[CM]A:

*Example 2 (*IND-CCA+EUF-CMA*).* Given an IND-CCA secure public key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ we define a new scheme $(\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ where

– $\mathcal{K}^*$: outputs the same public key $pk^* := pk$ as $\mathcal{K}$, but the secret key $sk^* := (sk, r^*)$ contains, in addition to the secret key sk determined by $\mathcal{K}$, a random bitstring r^* of length linear in the security parameter and such that r^* cannot occur as encryption of any plaintext;
– $\mathcal{E}^*$: identical with $\mathcal{E}$;
– $\mathcal{D}^*$: checks if the ciphertext is equal to r^*. If yes, the secret value sk is returned, otherwise the algorithm $\mathcal{D}$ is applied.

It is easy to see that $(\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ is secure in the sense of IND-CCA. Now suppose we are also given an EUF-CMA secure signature scheme $(\mathcal{K}, \mathcal{S}, \mathcal{V})$. Using the algorithm $\mathcal{K}^*$ just defined we form a new signature scheme $(\mathcal{K}^*, \mathcal{S}^*, \mathcal{V}^*)$ where

– $\mathcal{S}^*$: runs $\mathcal{S}$ to obtain a signature σ, and outputs (σ, r^*), i. e., the secret bitstring r^* is appended to each signature.

– $\mathcal{V}^*$: checks if a given signature (σ', r') satisfies $r' = r^*$, and if this holds outputs the same as $\mathcal{V}$ applied to σ'. If $r' \neq r^*$, then $\mathcal{V}^*$ outputs false.

Then $(\mathcal{K}^*, \mathcal{S}^*, \mathcal{V}^*)$ still offers security in the sense of EUF-CMA, but the combined public-key scheme $(\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*, \mathcal{S}^*, \mathcal{V}^*)$ is clearly not secure in the sense of IND-C[CM]A.

One would hope that problems as in the above (contrived) examples do not occur in a "natural" setting; however, it is not always easy to argue formally whether given encryption/signature schemes can be combined securely. In the next section, we give an example of a combination of two established schemes resulting in EUF-C[CM]A and IND-C[CM]A security in the random oracle model. A central proof tool is the *Forking Lemma* as used by Pointcheval and Stern in [21].

2.3 A secure combined public key scheme

We start by briefly describing the *Modified ElGamal (MEG)* signature scheme presented by Pointcheval and Stern in [21], and the encryption scheme of Fujisaki and Okamoto based on ElGamal encryption and one-time padding [14]. Moreover, we will summarize the essential results needed, to prove the resulting combined public key encryption scheme secure in the sense of Definition 2 and 3.

Fujisaki-Okamoto ElGamal based encryption For any $k \in \mathbb{N}$ and for any $q \in \mathbb{N}$ of length k, let $H_1 : \{0, \dots, q-1\} \longrightarrow \{0,1\}^k$ and $H_2 : \{0, \dots, q-1\} \times \{0,1\}^k \longrightarrow \{0, \dots, q-1\}$ be random oracles. The encryption scheme described in Figure 1 was presented in [14] as an instantiation of the general Fujisaki-Okamoto conversion using ElGamal encryption as asymmetric component and a one-time pad as symmetric component.

It follows from the results of [14], that the above scheme is IND-CCA secure in the random oracle model, provided that the Decisional Diffie-Hellman assumption holds for G.

Modified ElGamal signature scheme In [20] Pointcheval and Stern give a general strategy for providing security proofs for signature schemes in the random oracle model. One of the most prominent instantiations of this framework is a modification of the ElGamal Signature scheme (MEG), which we describe in Figure 2. At this, for any $k \in \mathbb{N}$ and any prime $p \in \mathbb{N}$ of length polynomial in k, let us assume a random oracle $F : \{0,1\}^* \times G \longrightarrow \{0, \dots, q-1\}$ is publicly known. The existential unforgeability of the MEG scheme was proven in [20, 21] as an application of the so-called *Forking Lemma*. We review here the basic results and ideas behind this proof, which will be needed in the sequel. For more details, we refer to the original papers.

- $\mathcal{K}$: on input k it
  - chooses a generator g of a cyclic group G of order q,
  - selects $x \in_R \{0, \ldots, q-1\}$ uniformly at random,
  - outputs (q, g, g^x) as public key and x as secret key;
- $\mathcal{E}$: on input $m \in \{0,1\}^k$, it
  - selects $r \in_R G$ uniformly at random,
  - computes $c_1 := r \cdot y^{H_2(r,m)}$, $c_2 := g^{H_2(r,m)}$ and $c_3 := H_1(r) \oplus m$,
  - outputs the ciphertext (c_1, c_2, c_3);
- $\mathcal{D}$: on input a ciphertext (c_1, c_2, c_3), it
  - computes $\hat{r} = c_1(c_2^x)^{-1}$,
  - retrieves $\hat{m} = H_1(\hat{r}) \oplus c_3$,
  - checks wether $c_1 = \hat{r} y^{H_2(\hat{r},\hat{m})}$, if this check fails, it outputs $\perp$, otherwise, it outputs $\hat{m}$.

Fig. 1. ElGamal encryption with Fujisaki-Okamoto conversion

Remark 2. In [20, 21], the case $q = p - 1$ for an α-hard prime p is considered. For our purposes, the case of q being prime is sufficient, as for the combination with the encryption scheme discussed in Section 2.3, we want to build on the Decisional Diffie Hellman assumption in the group G.

Pointcheval and Stern's results from [20, 21] apply to a large class of signature schemes, matching the following pattern: a signature of a message m consists of three components—a "commitment element" σ_1 sent by the signer, a random oracle image of this commitment together with the message, $h := F(m, \sigma_1)$, and a third component σ_2 which links σ_1 and h together and should be hard to compute without the secret signing key. These *generic signature schemes* can actually be obtained from any three-pass honest-verifier zero-knowledge identification protocol, as proven by Fiat and Shamir in [13].

For such generic schemes, it is proven [21, Theorem 10] that if a passive adversary is able to produce a valid forgery he will also be able to output two valid related signature tuples $(m, \sigma_1, h, \sigma_2)$, $(m, \sigma_1, h', \sigma_2')$, where h and h' are distinct and constructed using *different* random oracles F and F'. This result is commonly addressed as *Forking Lemma.* For the scenario considered here, this result has very strong implications, namely in [21, Theorem 18] the authors prove that for MEG, such two tuples provide a solution for the discrete logarithm problem on input (g, g^x), provided that p is an α-hard prime. Moreover, [21, Lemma 19] states that actually, for an α-hard prime p, the same reasoning can be applied in the case of an active adversary, as the signing oracle can be simulated without the secret key with an indistinguishable distribution. As a result, the existential unforgeability of the MEG scheme can be proven, in the random oracle model, under the discrete logarithm assumption.

– $\mathcal{K}$: on input k it
  - chooses a generator g of a subgroup G of $(\frac{\mathbb{Z}}{p\mathbb{Z}})^\times$ of order q. We thus represent the elements of G as integers mod p,
  - selects $x \in_R \{0, \ldots, q-1\}$ uniformly at random,
  - outputs (q, g, g^x) with $(g, g^x) \in G^2$ as public key and x as secret key;

– $\mathcal{S}$: on input a message $m \in \{0,1\}^*$, it
  - selects $K \in_R (\frac{\mathbb{Z}}{q\mathbb{Z}})^\times$ uniformly at random and defines $r := g^K \in G$,
  - sets $h := F(m, r)$ and selects $s \in \{0, \ldots, q-1\}$ so that $g^h = (g^x)^r r^s$, that is, s is the solution of the linear equation $F(m, r) = xr + Ks \mod q$,
  - outputs the triplet (r, h, s);

– $\mathcal{V}$: on input a triplet (r, h, s), it
  - outputs true if and only if $r \in G$ and $g^h = y^r r^s \mod p$.

Fig. 2. Pointcheval-Stern Modified ElGamal Signature

ElGamal based combined public key scheme Joining the above ElGamal based encryption and signature schemes in the natural way, we obtain a combined public key scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$, where

– $\mathcal{K}$: on input k, the key generation algorithm
  - chooses a random large α-hard prime p, of length polynomial in k (and larger than k),
  - chooses a generator g of a subgroup G of $(\frac{\mathbb{Z}}{p\mathbb{Z}})^\times$ of order q,
  - selects $x \in_R \{0, \ldots, q-1\}$ uniformly at random,
  - outputs (q, g, g^x) as public key and x as secret key;
– $\mathcal{E}, \mathcal{D}$: exactly as defined in Figure 1;
– $\mathcal{S}$: on input a message $m \in \{0,1\}^*$, the signing algorithm
  - selects $K \in_R (\frac{\mathbb{Z}}{q\mathbb{Z}})^\times$ uniformly at random and defines $r = g^K \in G$,
  - sets $h = F(m, r)$ and selects $s \in \{0, \ldots, q-1\}$ so that $g^h = (g^x)^r r^s$, that is, s is the solution of the linear equation $F(m, r) = xr + Ks \mod q$,
  - outputs the triple (r, h, s);
– $\mathcal{V}$: exactly as defined in Figure 2.

The following proposition establishes rather strong security guarantees for this combined public key scheme:

Proposition 1. *If the Decisional Diffie-Hellman assumption holds for G, the above ElGamal based combined public key scheme is* IND-C[CM]A *and* EUF-C[CM]A *secure in the random oracle model.*

Proof. We consider the attack scenarios described in Definitions 2 and 3: Assume the key generation algorithm has been executed and the corresponding public key (q, g, g^x) has been forwarded to the adversary $\mathcal{A}$. Thus, $\mathcal{A}$ has access to all public information and can execute the encryption algorithm of the Fujisaki-Okamoto public key encryption scheme and the verification algorithm of the MEG signature scheme. In addition, $\mathcal{A}$ has access to oracles $\mathcal{O}_\mathcal{D}$ and $\mathcal{O}_\mathcal{S}$ subject to the restrictions from Definition 2 respectively Definition 3.

EUF-C[C,M]A *security.* We start by arguing that $\mathcal{O}_\mathcal{S}$ is not needed by $\mathcal{A}$, with a similar argument as in [21, Lemma 19]. Let `SSim` be a simulator that on input any message m outputs a valid signature triple (σ_1, h, σ_2), such that `SSim`'s output distribution is indistinguishable from the output distribution of the signature algorithm. Such a simulator exists, as proven in [21, Lemma 19] for α-hard prime numbers. At this, it is important to note that `SSim` does not hold the secret signing key.

Our adversary $\mathcal{A}$ will only notice that he is interacting with `SSim` instead of $\mathcal{O}_\mathcal{S}$ provided that one the following events occurs:

- E_1: there exists a triple (r, h, s) output by the simulator on input a message m, so that at some point $\mathcal{A}$ submitted the query (m, r) to the random oracle F; or
- E_2: there exist two triples (r, h, s), (r, h', s'), with $h \neq h'$, output by the simulator when queried (twice) with the same input m.

Note that $\mathcal{A}$ will only derive distinguishing information from the encryption algorithm or from interacting with $\mathcal{O}_\mathcal{D}$ if at some point he queries the random oracle F; thus, this event is captured by event E_1 above.

Now, as argued in the proof of [21, Lemma 12], the probability of events E_1 and E_2 together is, up to a constant factor, bounded by the probability of success of $\mathcal{A}$ in making a forgery. As a result, we know that if $\mathcal{A}$ is able to create a forgery by querying $\mathcal{O}_\mathcal{S}$ with non-negligible probability, he will also be able to make a forgery interacting with `SSim` with non-negligible probability.

However, we can argue that $\mathcal{A}$ is not able to forge a signature with non-negligible probability interacting only with `SSim`: the *Forking Lemma* [21, Theorem 10] guarantees that such an adversary that is able to forge a signature with non-negligible probability, can also solve the underlying discrete logarithm problem in polynomial time. In other words, our adversary $\mathcal{A}$, interacting with `SSim` (which holds no secret key) and $\mathcal{O}_\mathcal{D}$ could decrypt arbitrary ciphertexts, which contradicts the IND-CCA security of the Fujisaki-Okamoto conversion. As a result, $\mathcal{A}$ cannot succeed when aiming at a forgery, and we have established EUF-C[CM]A security for the combined scheme.

IND-C[CM]A *security.* As we have argued that the signing oracle can be simulated without the secret key, our adversary is nothing more than a standard IND-CCA adversary against an IND-CCA secure scheme obtained with a Fujisaki-Okamoto conversion. Consequently, his advantage against this encryption scheme is negligible, and we see that the combined scheme is also IND-C[CM]A secure. □

3 Identity-based signature and encryption

In the context of identity-based public key cryptography, it appears natural to use the same identity for both encryption and signature purposes. One could explore the setting where a single key generation center is used for extracting signing and private decryption keys from user identities, and there is only a single set of public parameters; here we go one step further and consider, analogously as in the previous section, a situation in which each user has only one secret key to do both decrypting and signing.

3.1 Preliminaries and definitions

To adapt the definition of a *Combined Public Key Scheme* to the identity-based setting, we introduce the following definition:

Definition 4 (Combined identity-based public key scheme).
A combined identity-based public key scheme *is a tuple* $(\mathcal{I}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ *of polynomial time algorithms:*

- $\mathcal{I}$ *is a probabilistic* setup algorithm *that upon input the security parameter* 1^k *returns public system parameters* pk *and a secret master key* sk*.*
- $\mathcal{K}$ *is a probabilistic* key extraction algorithm *which upon input the public system parameters* pk*, the master key* sk*, and an identity* $id \in \{0,1\}^*$ *outputs a secret key* sk_{id}*, which can both be used for signature and decryption.*
- $\mathcal{E}$ *is a probabilistic* encryption algorithm *that upon input the public parameters* pk*, an identity* id *and a plaintext* m *computes a ciphertext* $c \leftarrow \mathcal{E}_{id,pk}(m)$*.*
- $\mathcal{D}$ *is a deterministic* decryption algorithm *that upon input a candidate ciphertext* c*, the public parameters* pk *and a secret key* sk_{id} *outputs a plaintext* $m \leftarrow \mathcal{D}_{sk_{id},pk}(c)$ *or an error symbol* $\perp$*.*
- $\mathcal{S}$ *is a probabilistic* signing algorithm *that upon input a message* m*, public parameters* pk *and a secret key* sk_{id} *outputs a signature* $\sigma \leftarrow \mathcal{S}_{sk_{id},pk}(m)$*.*
- $\mathcal{V}$ *is a deterministic* verification algorithm *that upon input the public parameters* pk*, an identity* id*, a message* m *and a candidate signature* σ *outputs* true *or* false*.*

For a pair (pk, sk) *generated by* $\mathcal{K}$ *we require that with overwhelming probability the obvious correctness condition holds for all private keys* sk_{id}*: For all messages* m *we have* $\mathcal{D}_{sk_{id},pk}(\mathcal{E}_{id,pk}(m)) = m$ *and*

$$\mathcal{V}_{id,pk}(m, \mathcal{S}_{sk_{id},pk}(m)) = \mathsf{true}.$$

To define the security of a combined identity-based public key scheme we adapt Definitions 2 and 3 accordingly, granting an adversary the (restricted) capability to obtain private keys, signatures and decryptions for identities of his choice:

Definition 5 (IND-ID-C[CM]A).
Let $(\mathcal{I}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ be a combined identity-based public key scheme, and let $\mathcal{A}$ be a probabilistic polynomial time adversary. Consider the following attack scenario:

1. *Compute a key pair $(pk, sk) \leftarrow \mathcal{I}(1^k)$, and hand pk as input to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ is given unrestricted access to a key extraction oracle $\mathcal{O}_{\mathcal{K}}$ to extract private keys, unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ and unrestricted access to a decryption oracle $\mathcal{O}_{\mathcal{D}}$.*[3] *At the end of this stage, $\mathcal{A}$ outputs two plaintexts $m_0 \neq m_1$ of equal length and an identity id_0 such that $\mathcal{O}_{\mathcal{K}}$ has not been queried for the corresponding secret key sk_{id_0}.*
3. *A value $b \in_R \{0, 1\}$ is chosen uniformly at random, and $\mathcal{A}$ learns a target ciphertext $c \leftarrow \mathcal{E}_{id_0,pk}(m_b)$.*
4. *The algorithm $\mathcal{A}$ is again given access to the key extraction oracle $\mathcal{O}_{\mathcal{K}}$, the signing oracle $\mathcal{O}_{\mathcal{S}}$, and the decryption oracle $\mathcal{O}_{\mathcal{D}}$, the only restrictions being that $\mathcal{O}_{\mathcal{D}}$ must not be queried to decrypt the target ciphertext c under sk_{id_0} and $\mathcal{O}_{\mathcal{K}}$ must not be queried for sk_{id_0}. At the end of this stage $\mathcal{A}$ outputs a guess b' for b.*

The advantage $\mathrm{Adv}_{\mathcal{A}} = \mathrm{Adv}_{\mathcal{A}}(k)$ *of $\mathcal{A}$ is defined as $|2 \cdot P[b = b'] - 1|$, and we call* $(\mathcal{I}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ secure in the sense of IND-ID-C[CM]A *if* $\mathrm{Adv}_{\mathcal{A}}$ *is negligible for all probabilistic polynomial time adversaries $\mathcal{A}$.*

Definition 6 (EUF-ID-C[CM]A).
Let $(\mathcal{I}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ be a combined identity-based public key scheme, and let $\mathcal{A}$ be a probabilistic polynomial time adversary. Consider the following attack scenario:

1. *Compute a key pair $(pk, sk) \leftarrow \mathcal{K}(1^k)$, and hand pk as input to $\mathcal{A}$.*
2. *The algorithm $\mathcal{A}$ is given unrestricted access to a key extraction oracle $\mathcal{O}_{\mathcal{K}}$ to extract private keys, unrestricted access to a signing oracle $\mathcal{O}_{\mathcal{S}}$ and unrestricted access to a decryption oracle $\mathcal{O}_{\mathcal{D}}$. At the end of this stage, $\mathcal{A}$ outputs a message m, an identity id_0 and a signature σ such that $\mathcal{O}_{\mathcal{S}}$ has not been queried for a signature on m under sk_{id_0} and such that $\mathcal{O}_{\mathcal{K}}$ has not been queried for sk_{id_0}.*

   The success probability $\mathrm{Succ}_{\mathcal{A}} = \mathrm{Succ}_{\mathcal{A}}(k)$ *of $\mathcal{A}$ is defined as*

$$P[\mathcal{V}_{id_0,pk}(m, \sigma) = \mathsf{true}],$$

and we call $(\mathcal{I}, \mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ secure in the sense of EUF-ID-C[CM]A *if* $\mathrm{Succ}_{\mathcal{A}}$ *is negligible for all probabilistic polynomial time adversaries $\mathcal{A}$.*

---

[3] Signing and decryption oracles can access the master key sk and the public parameters pk, and hence can answer signature and decryption queries for arbitrary identities; that is, queries for these oracles are of the form $(id, message)$. For a fixed identity id queries are replied using always the same secret key sk_{id} (cf. the INIT oracle in [2]).

3.2 A secure combined identity-based public key scheme

To obtain an example of a secure combined identity-based public key scheme in the above setting, we use an identity-based encryption scheme of Boneh and Franklin [7] and combine it with the identity-based signature scheme proposed by Hess in [17].

The FullIdent scheme of Boneh and Franklin In [7] an identity-based public key encryption scheme is proposed that can be considered as first practical proposal in this line of research and is referred to as FullIdent. It is similar to the IND-CCA secure ElGamal variation discussed in the previous section, in the sense that chosen ciphertext security is again obtained by means of the conversion technique of Fujisaki and Okamoto from [14]. For a detailed discussion of FullIdent and a proof of its IND-ID-CCA security we refer to [7, 15].

Consider $(G, +)$ and $(V, \cdot)$ two cyclic groups of prime order l, where $l = \Theta(2^k)$. Let $e : G \times G \to V$ be a suitable bilinear pairing. We write $G^* := G \setminus \{0\}$ and $V^* := V \setminus \{1\}$. Furthermore, consider four hash functions H_1, H_2, H_3, and H_4 of appropriate domain and range. With these ingredients, the scheme FullIdent is described in Figure 3.

Setup algorithm $\mathcal{I}$: chooses a random generator $P \in_R G$. Moreover, $Y_{\text{master}} := sk \cdot P$ is published, where $sk \in_R \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^\times$ is the uniformly at random chosen master key.
Key extraction $\mathcal{K}_{\mathbf{dec}}$: for an identity $id \in \{0,1\}^*$, the secret key is $sk_{id} := sk \cdot Q_{id}$, where $Q_{id} := H_1(id) \in G^*$.
Encryption algorithm $\mathcal{E}$: to encrypt a message $m \in \{0,1\}^n$ under the identity id, the following steps are performed:
- compute $Q_{id} := H_1(id) \in G^*$
- choose a random $\sigma \in_R \{0,1\}^n$ and set $r := H_3(\sigma, m) \in \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^\times$
- compute $g_{id} := e(Q_{id}, Y_{\text{master}}) \in V$

The ciphertext is $c := (r \cdot P, \sigma \oplus H_2(g_{id}^r), m \oplus H_4(\sigma))$.
Decryption algorithm $\mathcal{D}$: To decrypt a candidate ciphertext $c = (U, v, w)$ with secret key sk_{id}, the subsequent steps are performed:
- if $U \notin G^*$, the error symbol $\perp$ is returned
- compute $\sigma := v \oplus H_2(e(sk_{id}, U))$
- let $m := w \oplus H_4(\sigma)$ and $r := H_3(\sigma, m)$
- if $U \neq r \cdot P$ return the error symbol $\perp$, otherwise output m as decryption of c

Fig. 3. Identity-based encryption from [7]

Hess' identity-based signature scheme Consider, as above, $(G, +)$ and $(V, \cdot)$ two cyclic groups of prime order l, where $l = \Theta(2^k)$. Let $e : G \times G \to V$ be a suitable bilinear pairing. Furthermore, consider two hash functions

$$\begin{array}{ll} h : \{0,1\}^* \times V & \longrightarrow (\mathbb{Z}/l\mathbb{Z})^\times \text{ and} \\ H : \{0,1\}^* & \longrightarrow G^*. \end{array}$$

Setup algorithm $\mathcal{I}$: chooses a random generator $P \in_R G$. Moreover, $Y_{\text{master}} := sk \cdot P$ is published, where $sk \in_R \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^\times$ is the uniformly at random chosen master secret key.

Key extraction $\mathcal{K}_{\mathbf{sig}}$: for an identity $id \in \{0,1\}^*$, the secret key corresponding to this identity is computed from the secret master key by the issuing authority as $sk_{id} := sk \cdot H(id)$ and forwarded to the signer.

Signing algorithm $\mathcal{S}$: to sign a message m with sk_{id}, the signer chooses arbitrary $P_1 \in G^*$, picks a random integer $k \in \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^\times$ and computes:
- $r := e(P_1, P)^k$
- $v := h(m, r)$
- $u := v \cdot sk_{id} + k \cdot P_1$

The signature on m under sk_{id} then is $(u, v) \in G \times \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^\times$.

Verification algorithm $\mathcal{V}$: returns true if and only if a candidate signature (u, v) for a message m satisfies the equation $v = h(m, r)$, where r is computed as $r = e(u, P) \cdot e(H(id), -Y_{\text{master}})^v$.

Fig. 4. Identity-based signature scheme from [17]

The EUF-ID-CMA security of the scheme in Figure 4, in the random oracle model, relies on the hardness of the Computational Diffie Hellman Problem in G, as proven in [17, Theorem 1].

Combining FullIdent with Hess' signature scheme

The identity-based schemes just described both use a discrete logarithm sk of a public

$$Y_{\text{master}} = sk \cdot P$$

as secret master key. Moreover, also the key extraction algorithm is identical. Thus, it seems natural to form a combined identity-based public key scheme with the same setup and key extraction algorithm.

As public parameters pk we use a single value $Y_{\text{master}} = sk \cdot P \in G$ along with all the remaining (sk-independent) needed public parameters—like P and the specification of the random oracles $H = H_1, H_2, \ldots, H_4, h$—and of, only one, admissible bilinear map e. Note that both schemes will make use of the hash function H and of the bilinear map e; moreover, we shall assume that all these involved hash functions behave like independent random oracles.

Proposition 2. *In the random oracle model, if the Bilinear Diffie Hellman assumption for (G, e) holds, then the above combined identity-based public key scheme is secure in the sense of* IND-ID-C[CM]A *and in the sense of* EUF-ID-C[CM]A.

Proof. We show that a successful adversary in the sense of IND-ID-C[CM]A yields an IND-ID-CCA adversary against Boneh and Franklin's FullIdent scheme. Similarly, we argue that a successful EUF-ID-C[CM]A adversary would break the EUF-IND-CMA security of Hess' signature scheme.

IND-ID-C[CM]A *security.* Suppose we have an adversary $\mathcal{A}$ in the sense of IND-ID-C[CM]A violating ciphertext indistinguishability. We start by showing that the signing oracle $\mathcal{O}_S$ can be simulated for each identity without the corresponding secret key; thus, it is of no help to $\mathcal{A}$.

Consider a simulator `SSim` which on input of an identity id and a message m chooses $(u, v) \in_R G \times \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$ and defines the hash value $h(m, r) := v$ where $r = e(u, P) \cdot e(H(id), -Y_{\text{master}})^v$. The output of `SSim` is (u, v) which is a valid signature of m under id. The running time of `SSim` is comprised of the running time of the verification step (and of the book keeping for H and h) and is thus polynomial in the running time of $\mathcal{A}$.

This simulator `SSim` can only be distinguished from a true signing oracle if at some point a queried hash value $h(m, r)$ is already defined. The probability for this is $O(q_S^2/2^k)$ if at most most q_S signing queries are issued to `SSim`. Since $\mathcal{A}$ and thus q_S is polynomial in k, this probability is negligible.

From the above observations we see that an IND-ID-C[CM]A adversary $\mathcal{A}$ violating ciphertext indistinguishability can be transformed into an ordinary IND-ID-CCA attacker against the Boneh-Franklin scheme.

EUF-ID-C[CM]A *security.* Now suppose that an EUF-ID-C[CM]A adversary $\mathcal{A}$ successfully violates the existential unforgeability of the combined identity-based public key scheme in question.

Again, we argue that a decryption oracle can be simulated without the corresponding identities' private keys: such a simulator `DSim` exists for the full encryption scheme of [6]. A general argument for this is that the scheme arises as the Fujisaki-Okamoto transform of a γ-uniform and ID-OWE secure encryption scheme, see [6] and [14, Corollary 13]. In the following we give a more specific argument.

Valid ciphertexts are of the form

$$(rP, \sigma \oplus H_2(e(H(id), Y_{\text{master}})^r), m \oplus H_4(\sigma))$$

where H_2, H_3, H_4 are suitable fixed hash functions, σ, m are bitstrings and the equation $r = H_3(\sigma, m)$ holds true. Note that the map

$$f : (r, \sigma, m) \mapsto (rP, \sigma \oplus H_2(e(H(id), Y_{\text{master}})^r), m \oplus H_4(\sigma))$$

is bijective. The decryption function $\mathcal{D}$ inverts it internally to (σ, m) using the secret key, and returns m as the message if $r = H_3(\sigma, m)$ holds true, and the error symbol $\perp$ otherwise.

The simulator `DSim` is defined as follows. First, the simulator `DSim` is arranged to know the queries to the oracle of the hash function H_3. Second, for a decryption query on a given ciphertext c it checks whether one of the queries (σ, m) passed to the oracle of H_3 so far satisfies $f(r, \sigma, m) = c$ with $r = H_3(\sigma, m)$. If yes, the answer to the decryption query is m, otherwise the error symbol $\perp$.

We now compare two runs of $\mathcal{A}$ with identical inputs, in particular with identical random tapes and hence identical hash functions H, H_2, H_3, H_4, but in one case with access to the decryption function $\mathcal{D}$ and in the other case with access to the simulator `DSim`. Obviously $\mathcal{A}$ will execute identically until the, say, i-th decryption query will be answered differently by $\mathcal{D}$ and `DSim`. If `DSim` outputs the message m on input of c, then $r = H_3(\sigma, m)$ for r, σ, m with $f(r, \sigma, m) = c$, and $\mathcal{D}$ also outputs the message m. A difference can hence only occur if `DSim` outputs $\perp$ on input of c and $\mathcal{D}$ outputs a message on input of the same c. This means that, while again $r = H_3(\sigma, m)$ for r, σ, m with $f(r, \sigma, m) = c$ must hold, the oracle for H_3 has not been queried for σ, m in the run of $\mathcal{A}$ with `DSim`. Then the oracle for H_3 has also not been queried for σ, m in the run of $\mathcal{A}$ with $\mathcal{D}$, since the runs are identical up to the i-th decryption query (note that only queries to the oracle of H_3 outside `DSim` and $\mathcal{D}$ count). Randomizing over the input of $\mathcal{A}$ and the hash functions we see that the probability of different outputs of `DSim` and $\mathcal{D}$ in the i-th decryption query is equal to the probability that $\mathcal{A}$ computes c such that $r = H_3(\sigma, m)$ for r, σ, m with $f(r, \sigma, m) = c$ without querying the oracle of H_3 for σ, m. Since H_3 assumes random values in $(\mathbb{Z}/l\mathbb{Z})^\times$, this probability is $1/(l-1)$.

Now, if $\mathcal{A}$ makes at most q_D decryption queries, then the probability that `DSim` answers all these decryption queries like $\mathcal{D}$ is $(1 - 1/(l-1))^{q_D}$. Since $l = \Theta(2^k)$ and q_D is polynomial in k, this probability differs only negligibly from 1.

The running time of `DSim` is essentially that of encryption times the number of queries to the oracle of H_3 and is thus polynomial in k.

From the above observations we see that an EUF-ID-C[CM]A adversary $\mathcal{A}$ producing a forged signature can be transformed into an ordinary EUF-ID-CMA attacker against the scheme of Hess, which finishes the proof. $\square$

4 Conclusions

Building on earlier work in [16, 18, 11], our discussion offers formal security notions to analyze combined public key schemes both in an identity-based and in a non-identity-based setting. We give two concrete constructions using established public key schemes and prove them secure.

For the non identity-based case, the *Forking Lemma* turned out to be a powerful tool: as in our example the signing oracle can be simulated without knowledge of the secret key, a successful adversary constructing an existential forgery would be able to efficiently solve the mathematical problem underlying both the signature and the encryption scheme. Similarly, for the identity-based setting, the strategy is to argue that both the signing and decryption oracle can be simulated without the corresponding secret key. As a result, the "combined" adversary reduces to a standard one and the security level is thus inherited from the constituent encryption and signature schemes.

Aiming at the identification of further secure combined public key schemes, the above proof strategies appear to be quite promising.

Acknowledgment

The authors thank Nigel Smart for pointing out a relation of independent prior work by the authors which led to this joint paper.

References

1. J. H. An, Y. Dodis, and T. Rabin. On the Security of Joint Signature and Encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.
2. M. Bellare, C. Namprempere, and G. Neven. Security Proofs for Identity-Based Identification and Signature Schemes. Available at `http://www-cse.ucsd.edu/users/mihir/papers/ibi.html`, May 2004. Full version of [3].
3. M. Bellare, C. Namprempere, and G. Neven. Security Proofs for Identity-Based Identification and Signature Schemes. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer-Verlag, 2004.
4. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
5. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *Advances in Cryptology – Proceedings of EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.
6. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.
7. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. of Computing*, 32(3):586–615, 2003. Extended abstract appeared in [6].
8. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *42th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.
9. R. Canetti and T. Rabin. Universal Composition with Joint State. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 265–281. Springer-Verlag, 2003.
10. B. Chevallier-Mames, D. Hieu-Phan, and D. Pointcheval. Optimal Asymmetric Encryption and Signature Paddings. In J. Ioannidis, A. Keromytis, and M.Yung, editors, *Applied Cryptography and Network Security – Proceedings of ACNS 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 254–268. Springer-Verlag, 2005.
11. J.-S. Coron, M. Joye, D. Naccache, and P. Paillier. Universal Padding Schemes for RSA. In M. Yung, editor, *Advances in Cryptology – Proceedings of CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 226–241. Springer-Verlag, 2002.
12. Y. Dodis, M. J. Freedman, S. Jarecki, and S. Walfish. Versatile Padding Schemes for Joint Signature and Encryption. In *Proceedings of the 11th ACM Conference on Computer and Communications Security — CCS '04*, pages 344–353. ACM, 2004.

13. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology – Crypto '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987.
14. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In M. Wiener, editor, *Advances in Cryptology – CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer-Verlag, 1999.
15. D. Galindo. Boneh-Franklin Identity Based Encryption Revisited. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 791–802. Springer-Verlag, 2005.
16. S. Haber and B. Pinkas. Securely Combining Public-Key Cryptosystems. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 215–224. ACM, 2001.
17. F. Hess. Efficient Identity based Signature Schemes based on Pairings. In K. Nyberg and H. Heys, editors, *Proceedings of SAC 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer-Verlag, 2003.
18. Y. Komano and K. Ohta. Efficient Universal Padding Techniques for Multiplicative Trapdoor One-Way Permutation. In D. Boneh, editor, *Advances in Cryptology – Proceedings of CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 366–382. Springer-Verlag, 2003.
19. R. Nishimaki, Y. Manabe, and T. Okamoto. Universally Composable Identity-Based Encryption. In P.Q. Nguyen, editor, *Progress in Cryptology – VIETCRYPT 2006*, volume 4341 of *Lecture Notes in Computer Science*, pages 337–353. Springer-Verlag, 2006.
20. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.
21. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
22. Y. Zheng. Digital Signcryption or How to Achieve Cost(Signature & Encryption) $<<$ Cost(Signature) + Cost(Encryption). In B. S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.