Toward a Generic Construction of Convertible Undeniable Signatures from Pairing-Based Signatures

Laila El Aimani

b-it (Bonn-Aachen International Center for Information Technology),
Dahlmannstr. 2, D-53113 Bonn, Germany
elaimani@bit.uni-bonn.de

Abstract. Undeniable signatures were proposed to limit the verification property of ordinary digital signatures. In fact, the verification of such signatures cannot be attained without the help of the signer, via the confirmation/denial protocols. Later, the concept was refined to give the possibility of converting a *selected* signature into an ordinary one, or publishing a *universal* receipt that turns all undeniable signatures publicly verifiable.
In this paper, we present the first generic construction for convertible undeniable signatures from certain weakly secure cryptosystems and any secure digital signature scheme. Next, we give two specific approaches for building convertible undeniable signatures from a large class of pairing-based signatures. These methods find a nice and practical instantiation with known encryption and signature schemes. For instance, we achieve the most efficient undeniable signatures with regard to the signature length and cost, the underlying assumption and the security model. We believe these constructions could be an interesting starting point to develop more efficient schemes or give better security analyses of the existing ones.
Keywords: Undeniable signatures, Pairing-based signatures, Generic construction.

1 Introduction

Undeniable signatures were originally introduced in 1990 by Chaum and van Antwerpen [7] to limit the self-authenticating property of digital signatures. In fact, the verification algorithm in these signatures is replaced by a confirmation (denial) protocol between the verifier and the signer, in which the verifier learns the validity (invalidity) of the issued signature without being able to transfer his conviction to a third person. This cryptographic primitive proved valuable in many applications where privacy is a big concern, e.g., licensing software [7], electronic cash [6, 4, 14] and electronic voting and auctions.

In 1991, the notion of undeniable signature was boosted by Boyar et al. [3] to allow the conversion of a selected undeniable signature into an ordinary one by releasing a piece of information at a later time. The model supported also the universal conversion achieved by publishing a universal receipt (by the signer) that transforms all undeniable signatures into publicly verifiable ones.

1.1 Related work.

Since the introduction of undeniable signatures, a series of proposals sprang up, covering a variety of different aspects. Pairing-based signatures [1] have received a lot of attention in these settings. Actually, most such signatures include in the verification equation a pairing computation between a part of the signature and some other parameters. Therefore, if we implement the same signature in a non bilinear group, namely a group where the Decisional Diffie-Hellman problem (DDH) is intractable, the resulting signature cannot be publicly verifiable. Hence, the signer must perform a proof of equality/inequality of two discrete logarithms with the verifier. Such a duality between pairing-based signatures and undeniable signatures has been illustrated

[1] See Section 2 for definitions of pairings, bilinear groups, etc...

in the literature by some proposals, e.g., the BLS signatures [2] whose undeniable variant are the early Chaum and van Antwerpen [7] signatures or Boneh and Boyen's signatures [1] which resulted in Laguillaumie and Vergnaud's undeniable signatures [12]. All these signatures inherit the security properties of their underlying digital signatures and have their invisibility based on a variant of the DDH problem.
Unfortunately, this approach does not give the possibility of converting the resulting signatures. A tantalizing challenge is to propose a general approach that constructs undeniable signatures from (a large category of) pairing-based signatures with the possibility of converting them to ordinary ones.

1.2 Our contributions

We propose the first generic construction of convertible undeniable signatures from secure digital signatures and some weakly secure cryptosystems. Our design uses the "encryption of a signature" method [2] and relaxes the security requirement on the underlying cryptosystem, without compromising the overall security. As a consequence, we allow malleable cryptosystems in our design which impacts positively the efficiency of the confirmation/denial protocols.

Next, we give two efficient generic constructions of convertible undeniable signatures from a large class of pairing-based signatures. In fact, following the same principle, we shrink the set of signatures, upon which we build the undeniable signatures, down to a certain class of pairing-based signatures and we use appropriate encryption schemes. Our constructions find a very efficient instantiation and result in the most efficient convertible undeniable signature scheme without random oracles and whose security rests on standard assumptions.

2 Preliminaries

2.1 Bilinear maps

Definition 1. *Let $(\mathbb{G}, +)$ and $(\mathbb{H}, \times)$ be groups[3] of prime order d. Let P be a generator of $\mathbb{G}$. $\mathbb{G}$ is called a* bilinear group *if there exists a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$, with the following properties:*

1. *bilinearity: for all $(P, Q) \in \mathbb{G}^2$ and $a, b \in \mathbb{Z}_d$, $e(aP, bQ) = e(P, Q)^{ab}$,*
2. *efficient computability for any input pair, and*
3. *non-degeneracy: $e(P, P) \neq 1_{\mathbb{H}}$.*

2.2 Digital Signatures

A signature scheme Σ comprises three algorithms, keygen, sign, and verify:

- keygen is a probabilistic key generation algorithm which returns pairs of private and public keys $(\mathsf{sk}, \mathsf{pk})$ depending on the security parameter k,
- sign is a signing algorithm which takes on input a private key sk and a plaintext m and returns a signature σ, and
- verify is a deterministic algorithm which takes on input a public key pk, a signature σ and outputs 1 if the signature is valid and 0 otherwise.

Definition 2. *A signature scheme is said to be (t, ϵ, q_s)-EUF-CMA secure if no adversary $\mathcal{A}$, operating in time t and issuing at most q_s queries, wins the following game with probability greater than ϵ, where the probability is taken over all the random choices:*

[2] This method has been successfully used in a number of primitives such as designated confirmer signatures [5]. It consists in generating a signature on the message to be signed, then encrypting it. The validity or invalidity of the resulting signature are checked via concurrent proofs of knowledge.

[3] In the rest of the document, the group $\mathbb{G}$ is denoted additively whereas the group $\mathbb{H}$ is denoted multiplicatively.

Setup. *$\mathcal{A}$ is given the public parameters of the given signature scheme.*
Queries. *$\mathcal{A}$ queries the challenger for signatures on at most q_s messages.*
Output. *$\mathcal{A}$ outputs a pair (m, σ) and wins the game if m has not been queried before and* $\mathsf{verify}_{\mathsf{pk}}(m, \sigma) = 1$.

Remark 1. In case the adversary in the above game is allowed to output a previously queried message m along with a signature σ which was not obtained from the signing oracle, we call the signature scheme *strongly unforgeable against a chosen message attack* (SEUF-CMA secure).

2.3 Public-Key Encryption Schemes

An asymmetric encryption scheme comprises the following algorithms:

- keygen is a probabilistic key generation algorithm which returns pairs of private and public keys $(\mathsf{sk}, \mathsf{pk})$ depending on the security parameter k,
- $\mathsf{encrypt}$ is a probabilistic encryption algorithm which takes on input a public key pk and a plaintext m, and returns a ciphertext c, and
- $\mathsf{decrypt}$ is a deterministic decryption algorithm which takes on input a secret key sk and a ciphertext c, and returns the corresponding plaintext m or $\perp$.

A cryptosystem provides indistinguishability (IND) if it is difficult to distinguish pairs of ciphertexts based on the messages they encrypt. In case the adversary against the scheme has access to a decryption oracle, the scheme is said to be indistinguishable against chosen ciphertext attacks (IND-CCA), otherwise, if the adversary does not have access to any oracles, it is called indistinguishable against chosen plaintext attacks (IND-CPA). Formal definitions can be found in [15].

Finally, in some constructions which will follow later, we need the involved cryptosystem to meet the following property.

Definition 3. *A cryptosystem Γ is said to have* ***Property A*** *if the following holds. Let κ be a security parameter and let $(\mathsf{pk}, \mathsf{sk})$ be an output of $\Gamma.\mathsf{keygen}$. Consider the uniform distribution on the message space M considered by Γ. Then, the distribution, on the ciphertext space (which is only defined by the security parameter κ), corresponding to the random variable $\Gamma.\mathsf{encrypt}_{\mathsf{pk}}(m)$ ($m \in_R \mathsf{M}$) is computationally indistinguishable from uniform.*

2.4 Key/Data Encapsulation Mechanisms (KEM/DEM)

Key Encapsulation Mechanisms (KEMs). A KEM is a tuple of algorithms $\mathcal{K} = (\mathsf{keygen}, \mathsf{encap}, \mathsf{decap})$ where

- keygen probabilistically generates a key pair $(\mathsf{sk}, \mathsf{pk})$,
- encap, or the *encapsulation* algorithm which, on input a random nonce r and the public key pk, generates a *session key* denoted k and its *encapsulation* c, and
- decap, or the *decapsulation* algorithm. Given the private key sk and the element c, this algorithm computes the decapsulation k of c, or returns $\perp$ if c is invalid.

Definition 4. *A KEM is said to be (t, ϵ)-IND-CPA secure if no adversary $\mathcal{A}$, operating in time t, wins the following game with probability greater than ϵ:*

- ***Phase 1.*** *$\mathcal{A}$ gets the parameters of the KEM from his challenger.*
- ***Challenge.*** *The challenger computes a given encapsulation $c^\star$, then picks uniformly at random a bit b from $\{0, 1\}$. If $b = 1$, then he sets $k^\star$ to k_1 where $k_1 = \mathsf{decap}(c^\star)$. Otherwise, he sets $k^\star$ to a uniformly chosen string from the session keys space. The challenge is $(c^\star, k^\star)$.*
- ***Phase 2.*** *$\mathcal{A}$ outputs a bit b' (representing his guess of $k^\star$ being the decapsulation of $c^\star$) and wins the game if $b = b'$. We define $\mathcal{A}$'s advantage as $\mathsf{Adv}(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$, where the probability is taken over the random choices of the adversary $\mathcal{A}$ and of the challenger.*

Data Encapsulation Mechanisms (DEMs). A DEM is a secret key encryption scheme given by

- keygen, which probabilistically generates a key k,
- $\mathsf{encrypt}$, which on input a key k and a message m, outputs the ciphertext c, and
- $\mathsf{decrypt}$, which decrypts a given ciphertext c using the same key k used for encryption.

Definition 5. *A DEM $\mathcal{D}$ is said to be (t, ϵ)-INV-OT secure (invisible under a one time attack (INV-OT)) if no polynomial adversary $\mathcal{A}$, operating in time t, wins the following game with advantage greater than ϵ:*

- ***Phase 1.*** *The challenger runs the algorithm $\mathcal{D}.\mathsf{keygen}$ to obtain a key $\mathcal{D}.\mathsf{sk}$.*
- ***Challenge.*** *The adversary outputs eventually a message $m^\star$. The challenger picks uniformly at random a bit b from $\{0,1\}$. If $b = 0$, he encrypts $m^\star$, in $e^\star$, under $\mathcal{D}.\mathsf{sk}$. Otherwise, he chooses a string $e^\star$ uniformly at random from the ciphertext space.*
- ***Phase 2.*** *$\mathcal{A}$ outputs a bit b', representing his guess of $e^\star$ being the encryption of $m^\star$ and wins the game if $b = b'$. We define $\mathcal{A}$'s advantage as $\mathsf{Adv}(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$, where the probability is taken over the random choices of the adversary $\mathcal{A}$ and of the challenger.*

The hybrid encryption paradigm. It consists in combining KEMs with efficient Data Encapsulation Mechanisms (DEMs) to build encryption schemes. In fact, one can fix a session key k using the KEM, then uses it to encrypt a message using an efficient DEM. Decryption is achieved by first recovering the key from the encapsulation (part of the ciphertext) then applying the DEM decryption algorithm. It can be shown that one can obtain an IND-CPA cryptosystem from an IND-CPA KEM combined with a DEM indistinguishable under a one time attack (IND-OT). We refer to [10] for the necessary and sufficient conditions on KEMs and DEMs in order to obtain a certain level of security for the resulting hybrid encryption scheme.

3 Convertible Undeniable Signatures (CUS)

3.1 Definition

Setup. On input the security parameter κ, outputs the public parameters.

Key generation. Generates probabilistically a key pair $(\mathsf{sk}, \mathsf{pk})$.

Signature. On input the public parameters, the private key sk and a message m, outputs an undeniable signature μ.

Verification. This is an algorithm run by the signer, using his private key sk, to check the validity of an undeniable signature μ issued on m.

Confirmation/Denial protocol. These are interactive protocols between a prover and a verifier. Their common input consists of the public parameters of the scheme, the signature μ and the message m in question. The prover, that is the signer, uses his private key sk to convince the verifier of the validity (invalidity) of the signature μ on m.

Selective conversion. On input a message m and an alleged signature μ, output a string σ which can be universally verified as a valid digital signature on m if μ is a valid undeniable signature on m, otherwise output $\perp$.

Universal conversion. Releases a universal receipt, using sk, that makes all undeniable signatures universally verifiable.

Selective/Universal verification. On input a converted signature, a message, and the public key pk, output 1 if the signature is valid and 0 otherwise.

3.2 Security Model

In addition to completeness, soundness and non-transferability of the proofs inherent to the confirmation/denial protocols, a convertible undeniable signature scheme requires two further properties, that are unforgeability and invisibility.

Unforgeability. The natural security requirement that a convertible signature scheme should fulfill is the existential unforgeability against a chosen message attack (EUF-CMA). It is defined through the following game.

- **Setup.** The adversary $\mathcal{A}$ is given the public parameters of the scheme in addition to the universal receipt, which allows the universal conversion of all undeniable signatures.
- **Queries.** $\mathcal{A}$ queries the signing oracle adaptively on at most q_s messages. Note that there will be no need to query the confirmation/denial and selective conversion oracles since $\mathcal{A}$ has the universal receipt at his disposal.
- **Output.** At the end, $\mathcal{A}$ outputs a pair consisting of a message m, that has not been queried before, and a string μ. $\mathcal{A}$ wins the game if μ is a valid undeniable signature on m.

We say that a convertible undeniable signature scheme is (t, ϵ, q_s)-EUF-CMA secure if there is no adversary, operating in time t, that wins the above game with probability greater than ϵ.

Similarly, in case the adversary in the above game is allowed to output a previously queried message m along with a signature μ which was not obtained from the signing oracle, we call the undeniable signature scheme *strongly unforgeable against a chosen message attack* (SEUF-CMA secure).

Invisibility. Invisibility against a chosen message attack (INV-CMA) is defined through the following game between an attacker $\mathcal{A}$ and his challenger $\mathcal{R}$.

- $\mathcal{A}$ gets the parameters of the scheme from $\mathcal{R}$.
- **Phase 1.** $\mathcal{A}$ adaptively queries the signing, confirmation/denial and selective conversion oracles.
- **Challenge.** Eventually, $\mathcal{A}$ outputs a message $m^\star$ that has not been queried before to the signing oracle and requests a challenge signature $\mu^\star$. $\mathcal{R}$ picks a bit $b \in_R \{0, 1\}$. If $b = 1$, then $\mu^\star$ is generated as usual using the signing oracle, otherwise it is chosen uniformly at random from the signatures space.
- **Phase 2.** $\mathcal{A}$ can adaptively query the previous oracles with the exception of not querying $m^\star$ to the signing oracle nor $(m^\star, \mu^\star)$ to the verification or selective conversion oracles.
- **Output.** $\mathcal{A}$ outputs a bit b' representing his guess on $\mu^\star$ being a valid signature on $m^\star$. He wins the game if $b = b'$. We define $\mathcal{A}$'s advantage as $\mathsf{Adv}(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$, where the probability is taken over the random choices of the adversary $\mathcal{A}$ and of his challenger $\mathcal{R}$.

We say that a convertible undeniable signature scheme is $(t, \epsilon, q_s, q_v, q_{sc})$-INV-CMA secure if no adversary operating in time t, issuing q_s queries to the signing oracle, q_v queries to the confirmation/denial oracles and q_{sc} queries to the selective conversion oracle, wins the above game with advantage greater than ϵ.

4 A Systematic Approach for CUS from Digital Signatures and Certain Weakly Secure Cryptosystems

4.1 Design principle

We use the "encryption of a signature" method. Thus, we first generate a digital signature on the message to be signed, then encrypt the resulting signature using a suitable cryptosystem obtained from the hybrid encryption paradigm. Confirmation or denial of the resulting signatures

exist by virtue of Goldreich et al.'s result [9]. In fact, the verification and decryption algorithms in a signature scheme and a cryptosystem respectively define an NP (co-NP) language for which there exists a zero knowledge proof system.
This method has been in use for some time ago. For instance, Camenisch and Michels [5] used it for designated confirmer signatures. One of the main differences between the two proposals dwells in the security assumption on the cryptosystem. We actually require only IND-CPA secure KEMs and INV-OT secure DEMs (thus IND-CPA cryptosystems), versus IND-CCA cryptosystems. The consequences of this are twofold. First, we require a weak security notion on the cryptosystem without compromising the overall security. This gives many and simpler choices for the cryptosystem to be used. Second, we allow malleable cryptosystems in our construction, which impacts positively the confirmation/denial protocols efficiency. In fact, cryptosystems with homomorphic properties possess efficient decryption proofs of knowledge, i.e, one can prove efficiently the knowledge of the plaintext corresponding to a given ciphertext. Such schemes are not ruled out from our design.

4.2 Proposed construction

Let Σ be a digital signature scheme given by $\Sigma.\mathsf{keygen}$ which generates a key pair (private key $= \Sigma.\mathsf{sk}$, public key$= \Sigma.\mathsf{pk}$), $\Sigma.\mathsf{sign}$ and $\Sigma.\mathsf{verify}$.

Let furthermore Γ be a cryptosystem obtained using the hybrid encryption paradigm and described by $\Gamma.\mathsf{keygen}$ (that generates the pair (private key $= \Gamma.\mathsf{sk}$, public key$= \Gamma.\mathsf{pk}$)), $\Gamma.\mathsf{encrypt}$ and $\Gamma.\mathsf{decrypt}$. Note that the encapsulation of the key used to encrypt a given string is always contained in the ciphertext.

We assume for simplicity that the space of signatures produced by Σ is the same as the space of messages encrypted by Γ. Moreover, we define a special character, which we denote $\diamond$, that is not allowed to occur in a message to be signed. Finally, in the rest of the document, the symbol $\|$ will denote the operation which when applied to two strings m and c results in the "usual" concatenation of the string m, the special character $\diamond$ and the string c.

Let $m \in \{0,1\}^\star \backslash \{\diamond\}$ be a message, we propose the following scheme:

Setup. Invoke $\Gamma.\mathsf{setup}$ and $\Sigma.\mathsf{setup}$.

Key generation. Invoke $\Sigma.\mathsf{keygen}$ and $\Gamma.\mathsf{keygen}$ to generate $\Sigma.\mathsf{sk}$, $\Sigma.\mathsf{pk}$ $\Gamma.\mathsf{sk}$ and $\Gamma.\mathsf{pk}$. Set the public key to $(\Sigma.\mathsf{pk}, \Gamma.\mathsf{pk})$ and the private key to $(\Sigma.\mathsf{sk}, \Gamma.\mathsf{sk})$.

Signature. First compute an encapsulation c together with its decapsulation k using $\Gamma.\mathsf{pk}$. Then, compute a (digital) signature $\sigma = \Sigma.\mathsf{sign}_{\Sigma.\mathsf{sk}}(m\|c)$ on $m\|c$. Finally encrypt the resulting signature under $\Gamma.\mathsf{pk}$ (using k). Output $\mu = \Gamma.\mathsf{encrypt}_{\Gamma.\mathsf{pk}}(\sigma)$. Note that c is part of μ.

Verification (by the signer). To check the validity of an undeniable signature μ (that comprises the encapsulation c), issued on a certain message m, the signer first computes $\sigma = \Gamma.\mathsf{decrypt}_{\Gamma.\mathsf{sk}}(\mu)$, then calls $\Sigma.\mathsf{verify}$ on σ and $m\|c$ using $\Sigma.\mathsf{pk}$. μ is valid if and only if the output of the algorithm $\Sigma.\mathsf{verify}$ is 1.

Confirmation/Denial protocol. To confirm (deny) a purported signature μ (containing the encapsulation c) on a certain message m, the signer first computes $\sigma = \Gamma.\mathsf{decrypt}_{\Gamma.\mathsf{sk}}(\mu)$, then invokes the algorithm $\Sigma.\mathsf{verify}$ on σ and $m\|c$. According to the result, the signer issues a zero knowledge proof of knowledge of the decryption of μ that passes (does not pass) the verification algorithm $\Sigma.\mathsf{verify}$.

Selective conversion. To convert an alleged signature μ, the signer first checks its validity: if it is valid, he outputs $\Gamma.\mathsf{decrypt}_{\Gamma.\mathsf{sk}}(\mu)$, otherwise, he outputs $\perp$.

Universal conversion. Release $\Gamma.\mathsf{sk}$.

4.3 Security analysis

We first note that the properties of completeness, soundness and non-transferability of the confirmation/denial protocols are met by our construction as a direct consequence of the zero-knowledge proofs of knowledge. In the sequel, we prove that the construction resists existential forgeries and that signatures are invisible.

Theorem 1. *Our generic construction is (t, ϵ, q_s)-EUF-CMA secure if the underlying digital signature scheme is (t, ϵ, q_s)-EUF-CMA secure.*

Proof. Let $\mathcal{A}$ be an attacker that (t, ϵ, q_s)-EUF-CMA breaks the existential unforgeability of our construction. We will construct an adversary $\mathcal{R}$ that (t, ϵ, q_s)-EUF-CMA breaks the underlying digital signature scheme:

Key generation. $\mathcal{R}$ gets the parameters of the signature scheme in question from his challenger. Then he chooses an appropriate cryptosystem Γ (obtained from the hybrid encryption paradigm) with parameters $\Gamma.\mathsf{pk}$, $\Gamma.\mathsf{sk}$, $\Gamma.\mathsf{encrypt}$ and $\Gamma.\mathsf{decrypt}$. $\mathcal{R}$ fixes the above parameters as a setting for the undeniable signature scheme $\mathcal{A}$ is trying to attack.

Signature queries. For a signature query on a message m, $\mathcal{R}$ will first compute an encapsulation c together with its decapsulation k (using $\Gamma.\mathsf{pk}$). Then, he will request his challenger for a digital signature σ on $m\|c$. Finally, he will encrypt σ under $\Gamma.\mathsf{pk}$ (using k) and output the result to $\mathcal{A}$.

Final Output. Once $\mathcal{A}$ outputs his forgery $\mu^\star$ on $m^\star$, where $m^\star$ was never queried to the signing oracle, $\mathcal{R}$ will decrypt the signature to obtain $\sigma^\star$. If $\mu^\star$ is valid then by definition $\sigma^\star$ is a valid digital signature on $m^\star\|c^\star$, where $c^\star$ is the encapsulation of the key that was used to encrypt $\sigma^\star$. Suppose that there exists some $i \in [\![1, q_s]\!]$ such that $m^\star\|c^\star = m_i\|c_i$. Let j denotes the first occurence of the special symbol $\diamond$ in the strings $m^\star\|c^\star$ and $m_i\|c_i$. Since we assumed that the special character $\diamond$ does not occur in all the considered messages, for instance both m_i and $m^\star$, then equality of the strings $m^\star\|c^\star$ and $m_i\|c_i$ implies that the prefixes of those two strings up to the position j, namely $m^\star$ and m_i resp, are equal. This is a contradiction with the fact that $\mathcal{A}$ produced an existential forgery on the undeniable signature. We conclude then that $\mathcal{R}$ never requested his own challenger for a digital signature on $m^\star\|c^\star$. Therefore, the pair $(m^\star\|c^\star, \sigma^\star)$ forms a valid existential forgery on the underlying digital signature.

Note that there will be no need to simulate the confirmation/denial and selective conversion oracles since $\mathcal{A}$ has the universal receipt $\Gamma.\mathsf{sk}$ allowing the verification of the signatures.

□

The following remark is vital for the invisibility of the resulting undeniable signatures.

Remark 2. The previous theorem shows that existential unforgeability of the underlying digital signature scheme suffices to ensure existential unforgeability of the resulting construction. Actually, one can further show that this requirement on the digital signature (EUF-CMA security) guarantees also that no adversary, against the construction, can come up with a valid undeniable signature $\mu = (c, e)$ (c is the encapsulation used to generate the undeniable signature) on a message m that has been queried before to the signing oracle but where c was never used to generate answers (undeniable signatures) to the signature queries.

To prove this claim, we construct from such an adversary, say $\mathcal{A}$, an EUF-CMA adversary $\mathcal{R}$ against the underlying digital signature scheme, which runs in the same time and has the same advantage as $\mathcal{A}$. In fact, $\mathcal{R}$ will simulate $\mathcal{A}$'s environment in the same way described in the proof of Theorem 1. When $\mathcal{A}$ outputs his forgery $\mu^\star = (c^\star, e^\star)$ on a message m_i that has been previously queried to the signing oracle, $\mathcal{R}$ decrypts $\mu^\star$ in $\sigma^\star$, which by definition forms a

valid digital signature on $m_i \| c^\star$. Since by assumption $c^\star$ was never used to generate undeniable signatures on the queried messages, $\mathcal{R}$ never invoked his own challenger for a digital signature on $m_i \| c^\star$. Therefore, $\sigma^\star$ will form a valid existential forgery on the underlying signature scheme.

Theorem 2. *Our proposed construction is $(t, \epsilon, q_s, q_v, q_{sc})$-INV-CMA secure it uses a (t, ϵ', q_s)-EUF-CMA secure digital signature and a cryptosystem where the underlying KEM and DEM are $(t + q_s(q_v + q_{sc}), \epsilon \cdot (1 - \epsilon')^{q_v + q_{sc}})$-IND-CPA secure and INV-OT secure resp.*

Proof. Let $\mathcal{A}$ be an attacker that $(t, \epsilon, q_s, q_v, q_{sc})$-INV-CMA breaks our undeniable signatures, assumed to use a (t, ϵ', q_s)-EUF-CMA secure digital signature and an INV-OT secure DEM. We will construct an algorithm $\mathcal{R}$ that $(t + q_s(q_v + q_{sc}), \epsilon \cdot (1 - \epsilon')^{q_v + q_{sc}})$-IND-CPA breaks the underlying KEM.

Phase 1

Key generation. $\mathcal{R}$ gets the parameters of the KEM $\mathcal{K}$ from his challenger. Then he chooses an appropriate INV-OT secure DEM $\mathcal{D}$ together with a (t, ϵ', q_s)-EUF-CMA secure signature scheme Σ.

Signature queries. For a signature query on m. $\mathcal{R}$ first fixes a session key k together with its encapsulation c using $\mathcal{K}.pk$. Then, he computes a (digital) signature σ on $m \| c$ using $\Sigma.\mathsf{sk}$. Finally, he encrypts the produced signature (using k) and outputs the result to $\mathcal{A}$. $\mathcal{R}$ will maintain a list $\mathcal{L}$ of the queries he got (messages), the corresponding digital signatures and finally the encapsulations c and the keys k used to generate the undeniable signatures.

Verification (confirmation/denial) queries. For a signature $\mu = (c, e)$ on m, $\mathcal{R}$ will look up the list $\mathcal{L}$. If a record having as third component the encapsulation c, $\mathcal{R}$ will use the corresponding decapsulation k (fourth component of the record) to decrypt e in σ. If σ is a valid digital signature on $m \| c$, $\mathcal{R}$ will run the confirmation protocol, otherwise, he will run the denial protocol. $\mathcal{R}$ can issue such proofs of knowledge, without knowing the private key of $\mathcal{K}$, using the rewinding technique because the protocols are zero knowledge, thus simulatable.

This simulation differs from the real one when the signature $\mu = (c, e)$ on m is valid and c does not appear in any record of $\mathcal{L}$. We distinguish two cases: either m has never been queried to the signing oracle, then (m, μ) would correspond to an existential forgery on the undeniable signature scheme, which would lead to an existential forgery on the underlying signature scheme, by virtue of Theorem 1. The second case is when m has been previously queried to the signing oracle. According to Remark 2, this would lead to an existential forgery on the underlying digital scheme (since c was never used to generate an undeniable signature on the queried messages). Hence, the probability that both scenarios do not happen is at least $(1-\epsilon')^{q_v}$ because the underlying digital signature scheme is (t, ϵ', q_s)-EUF-CMA secure by assumption.

Selective conversion queries. For a selective conversion query on μ and m, $\mathcal{R}$ will proceed as he would do in a verification (confirmation/denial) query with the exception of outputting the decryption of μ instead of simulating the confirmation protocol, or the symbol $\perp$ instead of the denial protocol. Again the probability that this simulation does not differ from the real execution of the algorithm is at least $(1 - \epsilon')^{q_{sc}}$.

Challenge

Eventually, $\mathcal{A}$ outputs a challenge message $m^\star$. $\mathcal{R}$ will use his challenge $(c^\star, k^\star)$ to compute a digital signature using $\Sigma.\mathsf{sk}$ on $m^\star \| c^\star$. Then, he encrypts the resulting signature using $k^\star$ and outputs the result $\mu^\star$ to $\mathcal{A}$. Therefore, $\mu^\star$ is either a valid undeniable signature on $m^\star$ or an element indistinguishable from a random element in the (undeniable) signatures space ($k^\star$ is random according to Subsection 2.4 and the DEM is INV-OT secure). If $\mu^\star$ (in case

it is not a valid undeniable signature on m) is a random element in the undeniable signature space, then this complies with the game rules defined in Subsection 3.2. Otherwise, if it is only *indistinguishable from random*, then if the advantage of $\mathcal{A}$ is non-negligibly different from the advantage of an invisibility adversary in a real attack, then $\mathcal{A}$ can be easily turned into an attacker against the INV-OT security property of the DEM underlying the construction.

Bottom line is, under the INV-OT assumption of the DEM underlying the construction, the challenge undeniable signature $\mu^\star$ is either a valid undeniable signature on $m^\star$ or a random element in the signature space, which complies with the invisibility game in a real attack.

Phase 2

$\mathcal{A}$ will continue issuing queries to the signing, confirmation and denial oracles and $\mathcal{R}$ can answer as previously. Note that in this phase, $\mathcal{A}$ might request the verification or selective conversion of an undeniable signature $(c^\star, e_i)$ on a message $m_i \neq m^\star$. In this case, $\mathcal{R}$ will simply issue the denial protocol in case of a verification query, or the symbol $\perp$ in case of a selective conversion query. Following the same analysis as above, the probability that the simulation does not differ from the real execution is at least $(1-\epsilon')^{q_{sc}+q_v}$.

Final output

When $\mathcal{A}$ outputs his answer $b \in \{0,1\}$, $\mathcal{R}$ will forward this answer to his own challenger. Therefore $\mathcal{R}$ will $(t + q_s(q_v + q_{sc}), \epsilon \cdot (1-\epsilon')^{q_v+q_{sc}})$-IND-CPA break the KEM used in the construction.

□

5 An Efficient Transformation of Certain Pairing-Based Signatures to CUS

In the generic construction proposed in Section 4, the confirmation/denial protocols involve proofs of knowledge of the decryption of the undeniable signature which is a digital signature on some known data. Therefore, one needs to consider a set of cryptosystems and signatures for which such proofs could be performed efficiently. One solution to achieve this is to consider the following class of signatures.

5.1 Class $\mathbb{C}$ of signatures

Definition 6. $\mathbb{C}$ *is the set of pairing-based signatures such that:*

1. *The considered pairing e is from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{H}$.*
2. *A signature σ on a message m is written as $\sigma = (S, \bar{\sigma})$ such that*
   (a) *$\bar{\sigma} = \sigma \backslash S$ is independent of m and $(\mathsf{sk}, \mathsf{pk})$ the key pair related to the given signature scheme. I.e., there exists an algorithm that inputs a message m (from Σ's message space) and a key pair $(\Sigma.\mathsf{sk}, \Sigma.\mathsf{pk})$ (from the key pair space considered by Σ) and outputs a string statistically indistinguishable from $\bar{\sigma}$.*
   (b) *$S \in \mathbb{G}$ and the verification equation of the signature is of the form: $e(S, E) = f(\bar{\sigma}, m, PP)$, where E is some point in the group $\mathbb{G}$, f is a public function, m is the message in question and PP are the known public parameters of the signature scheme*

It is clear that this class of signatures captures a large category of pairing-based signatures. In fact, almost all (pairing-based) signatures [2, 1, 18, 17], that have been proposed so far, involve a pairing computation in the verification equation between the key-message-dependent part of the signature and other entities. Note that the key-message-independent part in [2, 18] is the empty string.

5.2 Proposed construction

Let Σ be a signature from $\mathbb{C}$ and Γ be an efficient decryption verifiable cryptosystem, i.e., a cryptosystem that accepts efficient zero knowledge proof of knowledge of the plaintext corresponding to a given ciphertext. Let further d denote the group order of $\mathbb{G}$ and p a suitable multiple of d such that Γ is IND-CPA secure in $\mathbb{Z}_p$ (the message space of Γ is included in $\mathbb{Z}_p$). Note that $p > d$ due the contrast of key sizes between finite-field (or ring) and elliptic-curve cryptography. Finally, we assume that Γ has Property A.

We devise a convertible undeniable signature scheme as follows. First, we choose $r \in_R \mathbb{Z}_p$ then encrypt it under Γ to result in $s = \Gamma.\mathsf{encrypt}_{\Gamma.\mathsf{pk}}(r)$. Next, generate a digital signature $(S, \bar{\sigma})$ on the message m to be signed concatenated with s. The signature consists of the triple $\mu = (s, (r \bmod d)S, \bar{\sigma})$. To confirm (deny) such a signature, the signer decrypts s then proves the equality (inequality) modulo d of the decryption of s and of the discrete logarithm of $e((r \bmod d)S, E)$ in base $f(\bar{\sigma}, m\|s, PP)$. Selective conversion is done by decrypting s in r, then recovering S from $(r \bmod d)S$ ($S = (r \bmod d)^{-1}(r \bmod d)S$); the converted signature consists of the pair $(S, \bar{\sigma})$. Finally, the universal conversion is achieved by releasing $\Gamma.\mathsf{sk}$.

Theorem 3. *Let $\mathcal{A}$ be a (t, ϵ, q_s)-EUF-CMA adversary against the above construction. Then, there exists a (t, ϵ, q_s)-EUF-CMA adversary against the underlying digital signature scheme.*

Proof. Consider the cryptosystem (obtained from the hybrid encryption paradigm) that maps an element of the form $(S, \bar{\sigma})$ to $(\Gamma.\mathsf{encrypt}(r), (r \bmod d)S, \bar{\sigma})$, where r is chosen uniformly at random from $\mathbb{Z}_p$, and apply Theorem 1. □

Remark 3. Remark 2 applies also here. That is, if an existential forger $\mathcal{A}$ against the construction is able to provide a forgery $(s = \Gamma.\mathsf{encrypt}(r), (r \bmod d)S, \bar{S})$ on a previously queried message m where s (and thus r) was never used to generate answers to $\mathcal{A}$'s signature queries, then one can construct an existential forger against the underlying signature scheme.

Theorem 4. *Our proposed construction is $(t, \epsilon, q_s, q_v, q_{sc})$-INV-CMA secure if uses a (t, ϵ', q_s)-EUF-CMA secure digital signature and a $(t + q_s(q_v + q_{sc}), \epsilon.(1 - \epsilon')^{q_v + q_{sc}}(1 - 1/d))$-IND-CPA secure cryptosystem which is assumed to have Property A.*

Proof. The proof strategy is very similar to the proof of Theorem 2. The only difference is the challenge phase and its analysis. The reduction $\mathcal{R}$ (against the cryptosystem) will output two messages $r_0, r_1 \in_R \mathbb{Z}_p$ to his challenger. We have $r_0 \neq r_1 \bmod d$ with probability at least $1 - 1/d$. $\mathcal{R}$ gets in response a ciphertext $s^\star$, which is the encryption of either r_0 or r_1. Let $m^\star$ be the challenge message output by the invisibility adversary $\mathcal{A}$. $\mathcal{R}$ generates the challenge undeniable signature as follows; he computes a digital signature $(S, \bar{\sigma})$ on $m^\star\|s^\star$, then he chooses a bit $b \in_R \{0, 1\}$ and finally outputs $\mu^\star = (s^\star, (r_b \bmod d)S, \bar{\sigma})$ to $\mathcal{A}$.

Let us analyze the conformity of this phase with the one in a real attack (as described in Paragraph 3.2). If $s^\star$ is a valid encryption of r_b, then $\mu^\star$ is a valid undeniable signature on $m^\star$. Otherwise, $s^\star$ is indistinguishable from a random element in the corresponding space, in the view of $\mathcal{A}$. In fact, both r_0 and r_1 are chosen uniformly at random from $\mathbb{Z}_p$, and Property A holds. Moreover, $(r_b \bmod d)$ is also random in $\mathbb{Z}_d$ (d is a divisor of p) and so is $(r_b \bmod d)S$ in $\mathbb{G}$ (the function that maps an element $r \in \mathbb{Z}_d$ to $rS \in \mathbb{G}$, for some $S \neq 0_{\mathbb{G}}$ is one-to-one). Finally, $\bar{\sigma}$ is statistically indistinguishable from a random element in the corresponding space, according to the properties of the signature scheme underlying the construction (Definition 6).

To sum up $\mu^\star = (s^\star, (r_b \bmod d)S, \bar{\sigma})$ is either a valid undeniable signature on $m^\star$, or a string indistinguishable from a random element in the signature space. Suppose that in the latter case, $\mu^\star$ is not a random element in the signature space and that $\mathcal{A}$ has an advantage which is non-negligeably different from the advantage of an invisibility adversary in a real attack, then

$\mathcal{R}$ can use $\mathcal{A}$ as a distinguisher for Property A of the cryptosystem Γ. We conclude then that, under the assumption that Γ has Property A, $\mu^\star$ is either a valid signature on $m^\star$ or a random string in the signature space, which complies with the invisibility game defined in Paragraph 3.2.

□

5.3 Confirmation/denial protocols in ElGamal-based constructions

In this section, we provide the confirmation/denial protocols for the undeniable signature resulting from combining a digital signature from the class $\mathbb{C}$ with the ElGamal encryption, in the way described above. In fact, according to [11], the textbook ElGamal encryption was shown to have Property A.

Let $\Sigma \in \mathbb{C}$ be an EUF-CMA signature scheme. If we combine Σ with the ElGamal cryptosystem in the way described in Subsection 5.2, then the obtained undeniable signature on a message m will be of the form $\mu = (g^t, ry^t, (r \bmod d)S, \bar{\sigma})$ where $(S, \bar{\sigma})$ is the digital signature on $m\|(g^t, ry^t)$ using Σ, and $(x, g^x = y)$ is the key pair related to ElGamal's cryptosystem implemented in the group $\langle g \rangle \subset (\mathbb{Z}_p^\times, \cdot)$, for some multiple p of d which has a large prime divisor. To convince a verifier of the validity (invalidity) of such a signature, the signer must prove that μ is in the following language:

$$\{\mu = (\mu_1, \mu_2, \mu_3, \mu_4) \mid \exists\, r \in \mathbb{Z}_p \colon \mathsf{DL}_g(y) = \mathsf{DL}_{\mu_1}(\mu_2 \cdot r^{-1}) \wedge e(\mu_3, E) = (\neq) f(\mu_4, m\|(\mu_1, \mu_2), PP)^{r \bmod d}\}$$

where $\mathsf{DL}_g(y)$ refers to the discrete logarithm of y in base g.
A zero knowledge proof of knowledge of this language has been proposed in [8] and proved sound, complete and zero knowledge.

1. The signer computes an encryption (z_1, z_2) of r, chooses $t' \in_R \mathbb{Z}_{|\langle g \rangle|}$ and $r' \in_R \mathbb{Z}_p^\times$ and sends $s' = (z_1 g^{t'}, z_2 r' y^{t'})$ and $\widetilde{\mu_3} = \mu_3^{r' \bmod d}$.
2. The verifier chooses $b \in \{0, 1\}$ and sends it to the signer.
3. If $b = 0$, then the signer sends back the pair (r', t'). Otherwise, the prover sends rr' and proves that s' is an encryption of rr' (this is a proof of equality of two discrete logarithms. See Figure 1).
4. If $b = 0$, the verifier checks that s' and $\widetilde{\mu_3}$ are computed as in Step 1. If $b = 1$, the verifier checks the proof of decryption of s'. If this fails, the verifier rejects the proof. Otherwise, if the signer is confirming a signature, the verifier accepts if $f(\mu_4, m\|(\mu_1, \mu_2), P)^{rr' \bmod d} = e(\widetilde{\mu_3}, E)$ and if the signer is denying a signature, the verifier accepts if $f(\mu_4, (\mu_1, \mu_2), PP)^{rr' \bmod d} \neq e(\widetilde{\mu_3}, E)$.

6 Toward a Generic Construction of CUS from Pairing-Based Signatures

In the generic construction proposed in Section 5, the considered cryptosystem operates on messages in $\mathbb{Z}_p$ for some suitable integer p. Therefore, the resulting undeniable signature might be long since it comprises an encryption using this cryptosystem. Since we consider digital signatures implemented in bilinear settings (the class $\mathbb{C}$), the proposed transformation to undeniable signatures fails to meet one of the main virtues of such signatures, namely achieve short signatures. In this section, we propose a remedy to this problem by defining a class of KEMs which lead to an efficient construction when combined with signatures from the class $\mathbb{C}$.

Prover		Verifier
		Choose $a, b \in_R \mathbb{Z}$
		Compute $c = e_1^a g^b$
	$\xleftarrow{c}$	
Choose $t \in_R \mathbb{Z}$		
Compute $h_1 = cg^t$		
Compute $h_2 = h_1^x$		
	$\xrightarrow{(h_1, h_2)}$	
	$\xleftarrow{(a, b)}$	
Verify that $c = e_1^a g^b$		
	$\xrightarrow{t}$	
		Verify that $h_1 = e_1^a g^{b+t}$ and $h_2 = (e_2 m^{-1})^a y^{b+t}$

Fig. 1. Proof system for $\{(e_1, e_2, m)\colon (e_1, e_2)$ is an ElGamal encryption of $m\}$ Common input: (e_1, e_2, m, y, g) and Private input: $x\colon y = g^x$

6.1 Defining the class $\mathbb{K}$ of KEMs

Definition 7. $\mathbb{K}$ *is the set of KEMs such that:*

1. *The KEM is implemented in a bilinear group* $\mathbb{G}$ *where the considered pairing* e *is from* $\mathbb{G} \times \mathbb{G}$ *to a group* $\mathbb{H}$.
2. *The session key space* K *is the group* $\mathbb{G}$.
3. *Let* k, E *be elements from* $\mathbb{G}$ *and* c *be a given encapsulation.*
   - *If* k *is the decapsulation* c*, then on the common input* $e(k, E)$*,* c *and* E*, there exists an efficient zero-knowledge proof* $\mathcal{C}$ *of this assertion (using the KEM private key),*
   - *otherwise, on the same common input* $e(k, E)$*,* c *and* E*, there exists an efficient zero-knowledge proof* $\mathcal{D}$ *of* k *not being the decapsulation of* c *(the proof always uses the KEM private key).*

A KEM in the class $\mathbb{K}$. We consider the following KEM that was first proposed in [1]:

- setup. Consider a bilinear group $\mathbb{G}$, with prime order d, generated by P.
- keygen. Probabilistically generate two secret values $x_1, x_2 \in \mathbb{Z}_d^\times$ and compute $X_1 = x_1 P$ and $X_2 = x_2 P$. Set the private key to $\mathsf{sk} = (x_1, x_2)$ and the public key to $\mathsf{pk} = (X_1, X_2)$.
- encap. On input a random nonce $(a, b) \in \mathbb{Z}_d^2$ and the public key pk, generate the *session key* $k = (a + b)P$ and its *encapsulation* $c = (aX_1, bX_2)$.
- decap. Given the private key sk and the element $c = (aX_1, bX_2)$, compute k as $k = x_1^{-1} aX_1 + x_2^{-1} bX_2$.

It is easy to see that indistinguishability against a chosen plaintext attack of this KEM (IND-CPA) rests on the *Decision Linear Assumption*:

Definition 8. ***Decision Linear Problem (DLP).*** *Given* U*,* V*,* H*,* aU*,* bV*,* $cH \in \mathbb{G}$*, output* 1 *if* $a + b = c \bmod (\#\mathbb{G})$ *and* 0 *otherwise.*

The traditional DDH problem (corresponding to $b = 0$) can be reduced to DLP. In fact, DLP is believed to be hard even in bilinear groups where DDH is easy.

Fact 1 *The linear Diffie-Hellman KEM described above is in the class* $\mathbb{K}$.

Proof. Let $c = (e_1, e_2) = (aX_1, bX_2)$ be a given encapsulation and $k, E \in \mathbb{G}$. If k is the decapsulation of c, then:

$$e(k, E) = e(aP, E)e(bP, E) = e(e_1, E)^{x_1^{-1}} e(e_2, E)^{x_2^{-1}}$$

Therefore, one can prove efficiently (using x_1 and x_2) that k is (not) the encapsulation of c by proving the knowledge of u, v such that $e(k, P) = (\neq)e(e_1, E)^u e(e_2, E)^v$ and $P = uX_1$ and $P = vX_2$. See Figures 2 and 3.

□

Prover		Verifier
$(k, \ell) \in_R [\![1, d-1]\!]^2$		
$C_1 = kX_1$		
$C_2 = \ell X_2$		
$c_3 = e(e_1, E)^k e(e_2, E)^\ell$		
	$\xrightarrow{C_1, C_2, c_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, d-1]\!]$
$c = k - ur \mod d$		
$d = \ell - vr \mod d$	$\xrightarrow{c, d}$	
		$C_1 \stackrel{?}{=} cX_1 + rP$
		$C_2 \stackrel{?}{=} dX_2 + rP$
		$c_3 \stackrel{?}{=} e(e_1, E)^c e(e_2, E)^\ell e(k, E)^r$

Fig. 2. Proof system for $\{(u, v)\colon e(k, E) = e(e_1, E)^u e(e_2, E)^v, P = uX_1, P = vX_2\}$ Common input: $(e(k, E), e_1, e_2, X_1, X_2, P, E)$, Private input: $(u, v)\colon P = uX_1, P = vX_2$

6.2 Construction

Following the notations in Subsection 6.1, we consider an EUF-CMA secure digital signature scheme $\Sigma \in \mathbb{C}$ and an IND-CPA secure KEM $\mathcal{K} \in \mathbb{K}$, where the considered groups are $\mathbb{G}$ and H. We assume that the proofs $\mathcal{C}$ and $\mathcal{D}$ are known to the signer. A convertible undeniable signature, on a given message m, can be obtained by first invoking $\mathcal{K}$ to fix a key k and its encapsulation c, then generating a digital signature $\sigma = (S, \bar{\sigma})$ on $m \| c$. The result is $\mu = (\mu_1, \mu_2, \mu_3) = (c, S + k, \bar{\sigma})$. In fact, the encryption algorithm of the considered DEM consists in adding the key to the message, whereas the decryption is the addition of the key inverse (in $\mathbb{G}$) to the ciphertext. Confirmation or denial of such a signature are achieved via the proofs $\mathcal{C}$ or $\mathcal{D}$ respectively, on the common input m, μ_1 and $e(\mu_2, E) f(\mu_3, m \| \mu_1, PP)^{-1}$. In fact, if $k = \mathcal{K}.\mathsf{decap}(\mu_1)$ and $e(k, E) = e(\mu_2, E) f(\mu_3, m \| c, PP)^{-1}$, then the signer issues $\mathcal{C}$ (using the private key of the KEM). Otherwise, if $k = \mathcal{K}.\mathsf{decap}(\mu_1)$ and $e(k, E) \neq e(\mu_2, E) f(\mu_3, m \| c, PP)^{-1}$, he issues the proof $\mathcal{D}$. Moreover, to convert a valid undeniable signature (μ_1, μ_2, μ_3), the signer uses the private key of $\mathcal{K}$ to decrypt (μ_1, μ_2) and outputs $(\mathcal{K}.\mathsf{decrypt}_{\mathcal{K}.\mathsf{sk}}(\mu_1, \mu_2), \mu_3)$. Finally, the universal conversion is achieved by releasing $\mathcal{K}.\mathsf{sk}$.

We discuss below the security analysis of the above construction.

Prover		Verifier
$(\ell, k_0, k_1, k_2) \in_R [\![1, q-1]\!]^4$ $c_0 = (e(k,E)e(e_1,E)^{-u}e(e_2,E)^{-v})^{\ell}$ $c_1 = e(k,E)^{-k_0}e(e_1,E)^{k_1}e(e_2,E)^{k_2}$ $C_2 = k_1X_1 - k_0P$; $C_3 = k_2X_2 - k_0P$		
	$\xrightarrow{c_0, c_1, C_2, C_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, d-1]\!]$
$c = k_0 - r\ell \mod d$ $d = k_1 - ur\ell \mod d$		
$e = k_2 - vr\ell \mod d$	$\xrightarrow{c, d, e}$	
		$c_0 \stackrel{?}{\neq} 1$
		$c_1 \stackrel{?}{=} e(k,E)^{-c}e(e_1,E)^{d}$ $\times e(e_2,E)^{e}c_0^{-r}$
		$C_2 \stackrel{?}{=} dX_1 - cP$
		$C_3 \stackrel{?}{=} eX_2 - cP$

Fig. 3. Proof system for $\{(u,v)\colon e(k,E) \neq e(e_1,E)^u e(e_2,E)^v, P = uX_1, P = vX_2\}$ Common input: $(e(k,E), e_1, e_2, X_1, X_2, P, E)$, Private input: $(u,v)\colon P = uX_1, P = vX_2$

Theorem 5. *The above construction is (t, ϵ, q_s)-EUF-CMA secure if so is the underlying signature scheme.*

Proof. Consider the cryptosystem that maps an element of the form $(S, \bar{\sigma})$ to $(c, k+S, \bar{\sigma})$ and apply Theorem 1. □

Remark 4. Remark 2 applies also here. That is, if an existential forger $\mathcal{A}$ against the construction is able to provide a forgery $(c, k+S, \bar{\sigma})$ on a previously queried message m where c (and thus k) was never used to generate answers to $\mathcal{A}$'s signature queries, then one can construct an existential forger against the underlying signature scheme.

Theorem 6. *Our proposed construction is (t, ϵ, q_s, q_v)-INV-CMA secure if it uses a (t, ϵ', q_s)-EUF-CMA secure signature scheme and a $(t + q_s q_v, \epsilon.(1-\epsilon')^{q_v})$-IND-CPA secure KEM.*

Proof. The proof strategy is very similar to the proof of Theorem 2. The only difference is the analysis of the challenge phase. The reduction $\mathcal{R}$ (against the KEM) plugs his challenge $(c^\star, k^\star)$ in the challenge of the invisibility adversary $\mathcal{A}$ as follows. Let $m^\star$ be the challenge message output by $\mathcal{A}$. $\mathcal{R}$ will first generate a digital signature $(S, \bar{\sigma})$ on $m^\star \| c^\star$. Then, he outputs $\mu^\star = (c^\star, k^\star + S, \bar{\sigma})$ as a challenge undeniable signature to $\mathcal{A}$.

Let us analyze the adequacy of this challenge signature with the one in a real attack (as described in Paragraph 3.2). In case $k^\star$ is indeed the decapsulation of $c^\star$, then $\mu^\star$ is a valid undeniable signature on $m^\star$. In the other case where $k^\star$ is a random element drawn from the key space of the considered KEM (the group $\mathbb{G}$), then $k^\star + S$ is also a random element in $\mathbb{G}$ (the function that maps $k \in G$ to $k + S \in \mathbb{G}$, for some $S \in \mathbb{G}$, is a permutation), finally $\bar{\sigma}$ is, by definition (of the class $\mathbb{C}$ of signatures), also (statistically indistinguishable from) a random element in the corresponding space. Therefore $\mu^\star$ corresponds to a random element in

the undeniable signature space, which conforms to the game rules of an invisibility adversary. The rest of the proof follows word-for-word from the proof of Theorem 2.

□

Instantiation of our framework with Waters' signatures [17] and the KEM described above results in a very efficient convertible undeniable signature scheme. In fact, the best scheme that was proposed so far [16] achieves the same security features (standard model and the same underlying standard assumptions). However, it has a longer signature and a higher signature generation and verification cost (approximately a multiplicative parameter k) and a higher key generation and universal conversion cost (a multiplicative parameter $2^{n/k}$), where k is a public parameter to be optimized and n is the length of the message to be signed.

7 Conclusion

In this paper, we proposed a construction for convertible undeniable signatures from secure digital signatures and some weakly secure cryptosystems. Next, we designed two efficient generic constructions for undeniable signatures from a large class of pairing-based signatures. These constructions found practical instantiations with some known signatures and cryptosystems. It might be good to analyze the security of the existing undeniable signature schemes or propose efficient ones using this technique. Finally, one is tempted to extend this approach to other "opaque" signatures such as directed signatures, or combine it with the techniques using commitment schemes in order to get better constructions.

8 Acknowledgments

I would like to thank the anonymous reviewers of IndoCrypt'08 for their helpful comments. Thanks go also to Joachim von zur Gathen and Le Trieu Phong for enlightening discussions. This work was supported by the B-IT Foundation and the Land Nordrhein-Westfalen. .

References

1. D. Boneh and X. Boyen, *Short Signatures Without Random Oracles.*, Advances in Cryptology - Eurocrypt 2004 (C. Cachin and J. Camenisch, eds.), LNCS, vol. 3027, Springer, 2004, pp. 56–73.
2. D. Boneh, B. Lynn, and H. Shacham, *Short Signatures from the Weil Pairing.*, J. Cryptology **17** (2004), no. 4, 297–319.
3. J. Boyar, D. Chaum, I. B. Damgård, and T. B. Pedersen, *Convertible undeniable signatures.*, Advances in Cryptology - Crypto'90 (A. J. Menezes and S. A. Vanstone, eds.), LNCS, vol. 537, Springer, 1991, pp. 189–205.
4. C. Boyd and E. Foo, *Off-line Fair Payment Protocols using Convertible Signatures.*, in Ohta and Pei [13], pp. 271–285.
5. J. Camenisch and M. Michels, *Confirmer Signature Schemes Secure against Adaptative Adversaries.*, Advances in Cryptology - Eurocrypt 2000 (B. Preneel, ed.), LNCS, vol. 1807, Springer, 2000, pp. 243–258.
6. D. Chaum and T. P. Pedersen, *Wallet Databases with Observers.*, Advances in Cryptology - Crypto'92 (E. F. Brickell, ed.), LNCS, vol. 740, Springer, 1993, pp. 89–105.
7. D. Chaum and H. van Antwerpen, *Undeniable Signatures.*, Advances in Cryptology - Crypto'89 (G. Brassard, ed.), LNCS, vol. 435, Springer, 1990, pp. 212–216.
8. I. B. Damgård and T. P. Pedersen, *New Convertible Undeniable Signature Schemes.*, Advances in Cryptology - Eurocrypt'96 (U. M. Maurer, ed.), LNCS, vol. 1070, Springer, 1996, pp. 372–386.
9. Oded Goldreich, Silvio Micali, and Avi Wigderson, *How to Prove all NP-Statements in Zero-Knowledge, and a Methodology of Cryptographic Protocol Design*, CRYPTO (A. M. Odlyzko, ed.), LNCS, vol. 263, Springer, 1986, pp. 171–185.
10. J. Herranz, D. Hofheinz, and E. Kiltz, *KEM/DEM: Necessary and Sufficient Conditions for secure Hybrid Encryption*, Available at http://eprint.iacr.org/2006/265.pdf, August 2006.

11. L. El Aimani, *Anonymity from public key encryption to undeniable signatures*, AFRICACRYPT 2009 (B. Preneel, ed.), LNCS, vol. 5580, Springer, 2009, pp. 217–234.
12. F. Laguillaumie and D. Vergnaud, *Short Undeniable Signatures Without Random Oracles: the Missing Link*, Progress in Cryptology - INDOCRYPT 2005 (S. Maitra, C. E. Veni Madhavan, and R. Venkatesan, eds.), LNCS, vol. 3797, Springer, 2005, pp. 283–296.
13. K. Ohta and D. Pei (eds.), *Advances in Cryptology* - ASIACRYPT *'98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings*, LNCS, vol. 1514, Springer, 1998.
14. D. Pointcheval, *Self-Scrambling Anonymizers.*, Financial Cryptography, 4th International Conference, FC 2000 (Y. Frankel, ed.), LNCS, vol. 1962, Springer, 2001, pp. 259–275.
15. Charles Rackoff and Daniel R. Simon, *Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack*, CRYPTO (J. Feigenbaum, ed.), LNCS, vol. 576, Springer, 1991, pp. 433–444.
16. Tsz Hon Yuen and Man Ho Au and Joseph K. Liu and Willy Susilo, *(Convertible) Undeniable Signatures Without Random Oracles*, ICICS (Sihan Qing and Hideki Imai and Guilin Wang, ed.), LNCS, vol. 4861, Springer, 2007, pp. 83–97.
17. B. Waters, *Efficient Identity-Based Encryption Without Random Oracles.*, Advances in Cryptology - EUROCRYPT 2005 (R. Cramer, ed.), LNCS, vol. 3494, Springer, 2005, pp. 114–127.
18. F. Zhang, R. Safavi-Naini, and W. Susilo, *An Efficient Signature Scheme from Bilinear Pairings and Its Applications.*, 7th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2004 (F. Bao, R. H. Deng, and J. Zhou, eds.), LNCS, vol. 2947, Springer, 2004, pp. 277–290.