Chaum's Designated Confirmer Signature Revisited

Jean Monnerat* and Serge Vaudenay

EPFL, Switzerland
http://lasecwww.epfl.ch

Abstract. This article revisits the original designated confirmer signature scheme of Chaum. Following the same spirit we naturally extend the Chaum's construction in a more general setting and analyze its security in a formal way. We prove its security in the random oracle model by using a random hash function and a random permutation. We notably consider the confirmer as an attacker against the existential forgery under an adaptive chosen-message attack. This security property is shown to rely on the hardness of forging signatures in a universal way of a classical existentially forgeable signature scheme. Furthermore, we show that the invisibility of the signatures under a non-adaptive chosen-message (aka lunchtime) attack relies on some invisibility properties of an existentially forgeable undeniable signature scheme. The existence of this cryptographic primitive is shown equivalent to the existence of public-key cryptosystems. It is also interesting to see that this article confirms the security of Chaum's scheme since our construction is a natural generalization of this scheme.

Key words: Designated confirmer signatures, random oracle model.

1 Introduction

Undeniable signatures [7] are some signature schemes which allow to authenticate a message in such a way that the recipient has to interact with the signer in order to be convinced of its validity. Otherwise the recipient cannot learn any information on the validity of the signature by its own. This kind of signature is useful for privacy protection when the signer would like to keep control on the spread of proofs of his signing. Some further applications such as the authenticity of software or auctions have been mentioned or presented in [5,8,9,15,25].

One drawback of such a signature scheme is that the physical integrity of the signer can be threatened to make him collaborate to the confirmation or denial protocol. This motivated Chaum in 1994 [6] to introduce designated confirmer signatures in which the ability to confirm/deny a signature is shifted to a delegate. The principal idea of this scheme is to mix an undeniable signature related to the confirmer with the hash of the message to be signed and then to

* Supported by a grant of the Swiss National Science Foundation, 200021-101453/1.

sign the result by using a classical existentially forgeable signature. In the same year, Okamoto [19] presented a generic construction based on some three move identification protocols and proved that the existence of confirmer signatures is equivalent to that of public-key cryptosystems. Since then, several new schemes have been proposed and some security issues have been explored [3,4,12,16].

The goal of this paper is to review the original scheme of Chaum [6] as well as the underlying ideas of his construction in a formal and more general setting. Namely, his original article neither presents a formal model nor a security proof. Our principal motivation is that the scheme of Chaum remains at this time one of the most simple and elegant construction of designated confirmer signature scheme. One motivation is to study the possibility to use an undeniable signature scheme in the construction of a designated confirmer signature, in particular reusing the confirmation and denial protocol.

As far as we know, the only generic constructions of designated confirmer signatures which are based on an undeniable signature scheme are that of Chaum [6] and the one of Okamoto [19]. The security of the latter was only proved in 2001 in [20] and its resistance against existential forgery under an adaptive chosen-message attack holds only against a classical adversary, i.e., anybody but the confirmer. To our best knowledge, the security of the Chaum's construction has not been proved yet. Moreover, the only known security flaw of this scheme is mentioned in [3]. The authors presented an attack against the invisibility of signatures in the adaptive scenario against the scheme of Michels and Stadler [16] and argued that the same kind of attack holds against the scheme of Chaum. In this attack, the attacker is able to transform a given message-signature pair in a new one such that the latter pair is valid only if the original pair is valid. Hence, the attacker breaks the invisibility of the first signature by sending the second pair to the confirmer for a confirmation (or denial) protocol.

Contributions of this paper. We extend the Chaum's construction based on an undeniable signature in a very natural way and formally study its security. To this end, we assume we have the two following cryptographic primitives at disposal: a classical existentially forgeable signature scheme and an existentially forgeable undeniable signature scheme. We then introduce the model of security and prove the security of this construction in the random oracle model. The main security result concerns the resistance against existential forgery under an adaptive chosen-message attack. This property is proved assuming that the underlying existentially forgeable signature scheme is resistant against a universal forgery under a no-message attack and holds even when the attacker is the confirmer. We furthermore show that the invisibility holds under a lunchtime chosen-message attack provided that the underlying undeniable signature scheme satisfies invisibility under a lunchtime known-message attack. This generalized Chaum construction does not satisfy invisibility against an adaptive attacker. We explain why this property certainly cannot be achieved without considerably changing the basic construction and its spirit. We also present a practical realization of this generalized Chaum construction. Finally, we dedicate a section of this paper to show that the existence of an existentially forgeable undeniable signature scheme which is invisible under a lunchtime known-message attack is equivalent to the existence of a public-key encryption scheme. This confirms that this construction is consistent with the result of Okamoto [19] and that depending on the required properties, an undeniable signature can lie in two classes of cryptographic primitives, those of public-key encryption and digital signatures.

2 Designated Confirmer Signature Scheme

We recall in this section the different algorithms of a designated confirmer signature scheme. In such a scheme we need to consider three entities that are the signer ($\mathbf{S}$), the confirmer ($\mathbf{C}$) and the verifier ($\mathbf{V}$). They all possess a pair of public/secret key $\mathcal{K}^{\mathbf{U}} := (\mathcal{K}_{\mathrm{p}}^{\mathbf{U}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{U}})$ for $\mathbf{U} \in \{\mathbf{S}, \mathbf{C}, \mathbf{V}\}$. The set of the message space is denoted by $\mathcal{M}$ and the set of the signature space is denoted by Σ. A designated confirmer signature is composed of the following algorithms.

Setup Let k be a security parameter. The setup is composed of three probabilistic polynomial time algorithms $\mathsf{Setup}^{\mathbf{U}}$ for $\mathbf{U} \in \{\mathbf{S}, \mathbf{C}, \mathbf{V}\}$ producing keys $\mathcal{K}^{\mathbf{U}} \leftarrow \mathsf{Setup}^{\mathbf{U}}(1^k)$. Furthermore, we assume that public keys are exchanged in an authenticated way.

Sign Let $m \in \mathcal{M}$ be a message. On the input of the signer's secret key $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}$ and confirmer's public key $\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}$, the (probabilistic) polynomial time algorithm Sign generates a signature $\sigma \leftarrow \mathsf{Sign}(m, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{p}}^{\mathbf{C}})$ of m (which lies in Σ).
We say that the pair (m, σ) is valid if there exists a random tape such that $\mathsf{Sign}(m, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{p}}^{\mathbf{C}})$ outputs σ. Otherwise, we say (m, σ) is invalid.

Confirm Let $(m, \sigma) \in \mathcal{M} \times \Sigma$ be a supposedly valid message-signature pair. $\mathsf{Confirm}$ is an interactive protocol between $\mathbf{C}$ and $\mathbf{V}$ i.e., a pair of interactive probabilistic polynomial time algorithms $\mathsf{Confirm}^{\mathbf{C}}$ and $\mathsf{Confirm}^{\mathbf{V}}$ such that m, σ, $\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}$ are input of both, $\mathcal{K}_{\mathrm{s}}^{\mathbf{C}}$ is the auxiliary input of $\mathsf{Confirm}^{\mathbf{C}}$ and $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$ is the auxiliary input of $\mathsf{Confirm}^{\mathbf{V}}$. At the end of the protocol, $\mathsf{Confirm}^{\mathbf{V}}$ outputs a boolean value which tells whether σ is accepted as a valid signature of m.

Deny Let $(m, \sigma') \in \mathcal{M} \times \Sigma$ be an alleged invalid message-signature pair. Deny is an interactive protocol between $\mathbf{C}$ and $\mathbf{V}$ i.e., a pair of interactive probabilistic polynomial time algorithms $\mathsf{Deny}^{\mathbf{C}}$ and $\mathsf{Deny}^{\mathbf{V}}$ such that m, σ', $\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$, $\mathcal{K}_{\mathrm{p}}^{\mathbf{V}}$ are input of both, $\mathcal{K}_{\mathrm{s}}^{\mathbf{C}}$ is the auxiliary input of $\mathsf{Deny}^{\mathbf{C}}$ and $\mathcal{K}_{\mathrm{s}}^{\mathbf{V}}$ is the auxiliary input of $\mathsf{Deny}^{\mathbf{V}}$. At the end of the protocol, $\mathsf{Deny}^{\mathbf{V}}$ outputs a boolean value which tells whether σ' is accepted as an invalid signature.

3 Security Requirements

Existential Forgery This notion protects the signer $\mathbf{S}$ from an attacker $\mathcal{A}$ which would like to forge a signature on a (possibly random) message $m \in \mathcal{M}$ without knowing the signer's secret key $\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}$. In this paper, we will consider the standard security notion of existential forgery under adaptive chosen-message attack defined by Goldwasser et al. [11] for classical digital signatures. We adapt this notion in our context as follows.

Definition 1. *The designated confirmer signature* Sign *is secure against an existential forgery under adaptive chosen-message attack if there exists no probabilistic polynomial time algorithm* $\mathcal{A}$ *which wins the following game with a non-negligible probability.*
Game: *$\mathcal{A}$ receives $\mathcal{K}_\mathrm{p}^\mathbf{C}$, $\mathcal{K}_\mathrm{p}^\mathbf{S}$, $\mathcal{K}_\mathrm{p}^\mathbf{V}$ (possibly $\mathcal{K}_\mathrm{s}^\mathbf{C}$) from $(\mathcal{K}_\mathrm{p}^\mathbf{C},\mathcal{K}_\mathrm{s}^\mathbf{C}) \leftarrow \mathsf{Setup}^\mathbf{C}(1^k)$, $(\mathcal{K}_\mathrm{p}^\mathbf{S},\mathcal{K}_\mathrm{s}^\mathbf{S}) \leftarrow \mathsf{Setup}^\mathbf{S}(1^k)$, $(\mathcal{K}_\mathrm{p}^\mathbf{V},\mathcal{K}_\mathrm{s}^\mathbf{V}) \leftarrow \mathsf{Setup}^{\boldsymbol{V}}(1^k)$, generated randomly and depending on a security parameter k. Then, $\mathcal{A}$ can query some chosen messages to a signing oracle, some chosen pairs $(m^*,\sigma^*) \in \mathcal{M} \times \Sigma$ to a confirmation (and denial) protocol oracle and interact with it in a confirmation (denial) protocol where the oracle plays the role of the prover. All these queries must be polynomially bounded in k and can be sent adaptively. $\mathcal{A}$ wins the game if it outputs a valid pair $(m,\sigma) \in \mathcal{M} \times \Sigma$ such that m was not queried to the signing oracle. We denote this probability of success by $\mathsf{Succ}^{\mathsf{ef-cma}}_{\mathsf{Sign},\mathcal{A}}(k)$.*

Invisibility of Signatures We present here a definition which is adapted from [3].

Definition 2. *We say that* Sign *satisfies the* invisibility property *under a lunch-time chosen (resp. known)-message attack if there exists no probabilistic polynomial time algorithm* $\mathcal{D}$ *called* invisibility distinguisher *which wins the following game with a non-negligible probability.*
Game: *$\mathcal{D}$ receives $\mathcal{K}_\mathrm{p}^\mathbf{C},\mathcal{K}_\mathrm{p}^\mathbf{S},\mathcal{K}_\mathrm{p}^\mathbf{V}$ (possibly $\mathcal{K}_\mathrm{s}^\mathbf{S}$) from $(\mathcal{K}_\mathrm{p}^\mathbf{C},\mathcal{K}_\mathrm{s}^\mathbf{C}) \leftarrow \mathsf{Setup}^\mathbf{C}(1^k)$, $(\mathcal{K}_\mathrm{p}^\mathbf{S},\mathcal{K}_\mathrm{s}^\mathbf{S}) \leftarrow \mathsf{Setup}^\mathbf{S}(1^k)$, $(\mathcal{K}_\mathrm{p}^\mathbf{V},\mathcal{K}_\mathrm{s}^\mathbf{V}) \leftarrow \mathsf{Setup}^{\boldsymbol{V}}(1^k)$. It can query some chosen messages to a signing oracle and some message-signature pairs $(m,\sigma) \in \mathcal{M} \times \Sigma$ to some oracles running the confirmation and denial protocol. After a given time (a lunch time), $\mathcal{D}$ does not have access to the oracles anymore. Then, it chooses two messages $m_0, m_1 \in \mathcal{M}$ and submits them to a challenger (resp. gets two messages $m_0, m_1 \in \mathcal{M}$ with uniform distribution). The challenger picks a random bit b. He sets $\sigma = \mathsf{Sign}(m_b,\mathcal{K}_\mathrm{s}^\mathbf{S},\mathcal{K}_\mathrm{p}^\mathbf{C})$. $\mathcal{D}$ receives σ. Finally, $\mathcal{D}$ outputs a guess bit b'. $\mathcal{D}$ wins the game if $b' = b$.
The advantage of such a distinguisher $\mathcal{D}$ is ε, where the probability that $b' = b$ is $\frac{1}{2} + \varepsilon$.*

Note that this definition is a little weaker than the definition of [3] in which $\mathcal{D}$ can continue to send queries to the oracles after the selection of m_0, m_1. We will discuss this point in Subsection 5.2.

Non-Coercibility This notion prevents that the signer **S** is coerced by anybody who would like to get a proof that a given signature was really generated by **S** after the signature is released. As far as the signer erases his intermediate computations, this notion can be regarded as an extension of the invisibility property in which the attacker is given $\mathcal{K}_\mathrm{s}^\mathbf{S}$. Indeed a signer who would keep in memory the random values needed to generate a signature could be coerced to prove later how this one was generated. Note also that we should distinguish the non-coercibility from the receipt-freeness where the signer would be unable to keep a proof that he really generated a given signature even if he meant to. This extends the non-coercibility to the non-corruptibility.

As additional security properties related to the confirmation and denial protocols, we have the *completeness*, the *soundness* and the *non-transferability.* The completeness ensures that a protocol always passes when the prover and the verifier follow it correctly. The soundness of the confirmation (resp. denial) protocol prevents from a malicious prover to prove that an invalid (resp. valid) signature is valid (resp. invalid). The *non-transferability* of the confirmation (resp. denial) protocol prevents a verifier from transferring the proof of the validity (resp. invalidity) of a signature to any third party. This concept was first stated in [14]. Moreover, a generic construction based on trapdoor commitments [2] is also given in this article. Formal definitions of these notions are given in [3].

4 The Generalized Chaum's Construction

4.1 Building Blocks

Existentially Forgeable Signature We consider an existentially forgeable signature ExSign such as the plain RSA or plain DSA[1] scheme. We have a setup which generates the keys associated to this scheme (that of $\mathbf{S}$), $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ which depends on a security parameter k. Let $\mathcal{M}_{\mathrm{ex}}$ denote the message space and Σ_{ex} denote the signature space of this scheme. We have

$$\sigma_{\mathrm{ex}} \leftarrow \mathsf{ExSign}_{\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}}(m_{\mathrm{ex}}), \quad 0 \text{ or } 1 \leftarrow \mathsf{ExVerify}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}}(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}})$$

depending on whether $(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}) \in \mathcal{M}_{\mathrm{ex}} \times \Sigma_{\mathrm{ex}}$ is a valid message-signature pair. We also have a probabilistic algorithm $(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}) \leftarrow \mathsf{ExForge}(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}})$ which existentially forges a valid message-signature pair such that m_{ex} is *uniformly distributed* in $\mathcal{M}_{\mathrm{ex}}$.

For proving the security of Sign, we will need to assume that ExSign satisfies universal unforgeability under a no-message attack.

Definition 3. *We say that the signature scheme* ExSign *resists against a universal forgery under a no-message attack if there exists no probabilistic polynomial time algorithm $\mathcal{B}$ that wins the following game with a non-negligible probability.*
Game: *$\mathcal{B}$ first receives the public key $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$ from* $(\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ *generated randomly and depending on the security parameter k. Then, $\mathcal{B}$ receives a challenged message $m_{\mathrm{ex}} \in \mathcal{M}_{\mathrm{ex}}$ which is uniformly picked at random. At the end, $\mathcal{B}$ wins this game if it outputs a signature σ_{ex} such that* $\mathsf{ExVerify}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}}(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}) = 1$.

Our definition of universal forgery is slightly weaker than usual as in [22], where a successful attacker should be able to forge a valid signature to every challenged message of the message space. In many situations such as plain RSA or plain DSA where messages can be blinded, the two notions are equivalent.

[1] Plain DSA is DSA without a hash function.

Group Structure We need $\mathcal{M}_{\text{ex}}$ to form *a group* with an internal operation $\odot$. The inverse of an element $m_{\text{ex}} \in \mathcal{M}_{\text{ex}}$ with respect to this group operation is simply denoted m_{ex}^{-1}.

Existentially Forgeable Undeniable Signature We consider an existentially forgeable undeniable signature scheme UnSign whose associated pair of keys is that of $\mathbf{C}$ i.e. $(\mathcal{K}_{\text{p}}^{\mathbf{C}}, \mathcal{K}_{\text{s}}^{\mathbf{C}}) \leftarrow \mathsf{Setup}^{\mathbf{C}}(1^k)$. We denote the message space $\mathcal{M}_{\text{un}}$ and the signature space Σ_{un}. We have two probabilistic polynomial time algorithms

$$\sigma_{\text{un}} \leftarrow \mathsf{UnSign}(\mathcal{K}_{\text{s}}^{\mathbf{C}}, m_{\text{un}}) \text{ and } (m_{\text{un}}, \sigma_{\text{un}}) \leftarrow \mathsf{UnForge}(\mathcal{K}_{\text{p}}^{\mathbf{C}}),$$

where the latter outputs a valid message-signature pair such that m_{un} is uniformly distributed. Furthermore, we also have two interactive protocols $\mathsf{UnConfirm}$ and UnDeny between $\mathbf{C}$ and $\mathbf{V}$. The properties are the same as for the algorithms $\mathsf{Confirm}$ and Deny.
We will assume that the function $\mathsf{UnSign}(\mathcal{K}_{\text{s}}^{\mathbf{C}}, \cdot)$ *is balanced on the set* Σ_{un} for any secret key $\mathcal{K}_{\text{s}}^{\mathbf{C}}$. So, the probability for a pair $(m_{\text{un}}, \sigma_{\text{un}})$ uniformly picked at random in $\mathcal{M}_{\text{un}} \times \Sigma_{\text{un}}$ to be valid is equal to $\nu := v/|\Sigma_{\text{un}}|$, where v denotes the number of valid signatures related (and independent) to each m_{un}.

Some examples of such undeniable signatures are the MOVA scheme [17], the RSA based scheme from [9], the scheme of Chaum [5] based on the discrete logarithm problem and the generic scheme [18] based on group homomorphisms. All these schemes present this property provided that we remove some hash functions or pseudorandom generators. Furthermore, we note that these obtained signatures schemes are deterministic and therefore cannot satisfy the invisibility property under a chosen-message attack.

Random Hash Function We consider a hash function $h : \mathcal{M} \to \mathcal{M}_{\text{ex}}$ which is collision-resistant. We furthermore assume that h is *full-domain* i.e., its range is the full set $\mathcal{M}_{\text{ex}}$. h will be considered as a random oracle.

Random Permutation We consider a public permutation $C : \mathcal{M}_{\text{ex}} \to \mathcal{M}_{\text{ex}}$. C will be considered as a random permutation oracle (see [21,23]) i.e., C is picked uniformly at random among all permutations over $\mathcal{M}_{\text{ex}}$. We assume that we can send queries to the oracle C and the oracle C^{-1}.

Representation Function We consider a fixed bijection $B : \mathcal{M}_{\text{un}} \times \Sigma_{\text{un}} \to \mathcal{M}_{\text{ex}}$. In what follows, we will always work with the function $\mathcal{F} := C \circ B$ instead of C and B separately. Note that $\mathcal{F}$ is then a random bijective function.

4.2 The Scheme

The generic construction we proposed is a natural generalization of Chaum's scheme [6]. The signer generates a valid message-signature pair with respect to an existentially forgeable undeniable signature scheme. Then the signer mixes this pair with a message digest of the message and finally signs the result in a classical way using ExSign. The validity of this designated confirmer signature will then rely on the validity of the message-signature pair which can only be confirmed by the confirmer. Since ExSign is existentially forgeable, anybody could have produced a signature with an invalid message-signature pair. On the other hand, when the message-signature pair is valid the designated confirmer signature can be produced only by the signer. So, without the help of the confirmer it is not possible to deduce the validity or invalidity of a designated confirmer signature.

Setup Three pairs of keys are generated $(\mathcal{K}_{\mathrm{p}}^{\mathbf{U}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{U}}) \leftarrow \mathsf{Setup}^{\mathbf{U}}(1^k)$ from a security parameter k, where $\mathbf{U} \in \{\mathbf{S}, \mathbf{C}, \mathbf{V}\}$.

Sign Let $m \in \mathcal{M}$ be a given message to sign. The signer runs the algorithm UnForge to obtain a pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ and computes $h(m)$. He then computes $m_{\mathrm{ex}} := \mathcal{F}(m_{\mathrm{un}}, \sigma_{\mathrm{un}}) \odot h(m)$. The designated confirmer signature of m is then $\sigma = (m_{\mathrm{ex}}, \sigma_{\mathrm{ex}})$, where $\sigma_{\mathrm{ex}} \leftarrow \mathsf{ExSign}_{\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}}(m_{\mathrm{ex}})$.

Confirm The verifier and the confirmer check that $\mathsf{ExVerify}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}}(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}) = 1$. Then, they compute $m_{\mathrm{ex}} \odot h(m)^{-1}$, apply $\mathcal{F}^{-1}$, and retrieve $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$. Then $\mathbf{V}$ interacts with $\mathbf{C}$ in a proof protocol in which $\mathbf{C}$ proves that $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ is valid using UnConfirm. If this is verified the protocol outputs 1.

Deny In the denial protocol, the verifier and the confirmer first check that $\mathsf{ExVerify}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}}(m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}) = 1$ and then retrieve $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ as in the confirmation. Then $\mathbf{V}$ interacts with $\mathbf{C}$ in a proof protocol in which $\mathbf{C}$ proves that $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ is invalid using UnDeny. If this is verified the protocol outputs 1.

Note that the confirmer could also confirm or deny signatures in an anonymous way: he does not need σ_{ex} nor m_{ex} but only m_{un} and σ_{un} which contain no information about the signer or the message. This could be suitable for some applications.

5 Security Results

5.1 Security against Adaptive Chosen-Message Existential Forgeries

Theorem 4. *The scheme* Sign *resists against existential forgery under an adaptive chosen-message attack provided that*

1. *h is a random hash function oracle and C/C^{-1} is a random permutation oracle*
2. ExSign *resists against universal forgery under a no-message attack*
3. *valid $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ pairs are sparse in $\mathcal{M}_{\mathrm{un}} \times \Sigma_{\mathrm{un}}$ (i.e. $\nu \ll 1$)*

*even if the attacker is the confirmer **C**.*
More precisely, for any attacker $\mathcal{A}$ which wins in the game of existential forgery under an adaptive chosen-message attack against Sign *with success probability* $\mathsf{Succ}^{\mathsf{ef-cma}}_{\mathsf{Sign},\mathcal{A}}(k) = \varepsilon$ *using q_h h-queries, $q_{\mathcal{F}}$ $\mathcal{F}$-queries, $q^*_{\mathcal{F}}$ $\mathcal{F}^{-1}$-queries, and q_{S}* Sign *queries, we can construct another attacker $\mathcal{B}$ which wins the game of universal forgery under a no-message attack against* ExSign *with success probability*

$$\Pr[\mathsf{Succ}^{\mathsf{uf-nma}}_{\mathsf{ExSign},\mathcal{B}}(k)] \geq \frac{1}{q_{\mathcal{F}} \cdot q_h} \left(\varepsilon - \frac{(q_{\mathcal{F}} + q^*_{\mathcal{F}})^2}{|\mathcal{M}_{\mathrm{ex}}|} - 2\nu \right)$$

using one run of $\mathcal{A}$.

Proof. For this proof, following Shoup's methodology [26], we will provide a sequence of games beginning from the real attack and reach a game allowing to deduce a universal forgery against ExSign. $\mathcal{B}$ is given a challenged public key $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$ and a challenged message $m_{\mathrm{chal}} \in \mathcal{M}_{\mathrm{ex}}$ for which it has to forge a signature σ_{chal} such that $\mathsf{ExVerify}_{\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}}(m_{\mathrm{chal}}, \sigma_{\mathrm{chal}})$ outputs 1 with a non-negligible probability.

Game 1. Here, we consider the real attack game with the random oracle h and random function oracle $\mathcal{F}$. First, $\mathcal{A}$ receives a challenged public key uniformly picked at random $\mathcal{K}_{\mathrm{p}}^{\mathbf{S}}$ for which it will have to output an existential forgery. Since the attacker $\mathcal{A}$ can be the confirmer, $\mathcal{A}$ gets also the confirmer key pair $(\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{C}})$. Note that it can simulate $\mathsf{Confirm}^{\mathbf{C}}$ and $\mathsf{Deny}^{\mathbf{C}}$, so we do not need to give $\mathcal{A}$ an access to the denial and confirmation protocol. The attacker makes adaptively and in any order the following queries:

- $\mathcal{A}$ sends q_h messages $m_1, \ldots, m_{q_h} \in \mathcal{M}$ to the random oracle h and receives the corresponding hash values $h_1, \ldots, h_{q_h}$.
- $\mathcal{A}$ sends $q_{\mathcal{F}}$ pairs $(m_{\mathrm{un},1}, \sigma_{\mathrm{un},1}), \ldots, (m_{\mathrm{un},q_{\mathcal{F}}}, \sigma_{\mathrm{un},q_{\mathcal{F}}})$ to the random function oracle $\mathcal{F}$ and receives the corresponding values $f_1, \ldots, f_{q_{\mathcal{F}}}$.
- $\mathcal{A}$ sends $q_{\mathcal{F}}^*$ elements $f_1^*, \ldots, f_{q_{\mathcal{F}}^*}^*$ to the random function oracle $\mathcal{F}^{-1}$ and receives the corresponding values $(m_{\mathrm{un},1}^*, \sigma_{\mathrm{un},1}^*), \ldots, (m_{\mathrm{un},q_{\mathcal{F}}^*}^*, \sigma_{\mathrm{un},q_{\mathcal{F}}^*}^*)$.
- $\mathcal{A}$ sends q_{S} messages $m_1^{\mathrm{s}}, \ldots, m_{q_{\mathrm{S}}}^{\mathrm{s}}$ to the signing oracle Sign (with respect to the challenged public key) and receives the corresponding signatures $\sigma_1, \ldots, \sigma_{q_{\mathrm{S}}}$. We assume that q_h and $q_{\mathcal{F}}$ includes the queries made by Sign.

After these queries, $\mathcal{A}$ outputs a message m (not queried to the signing oracle) with a correct forged signature σ with success probability $\Pr[\mathsf{S}_1] = \varepsilon$. In what follows, we denote the probability event that $\mathcal{A}$ succeeds in the **Game i** as S_i.

Note that the challenged public key $\mathcal{B}$ received in the universal forgery game against ExSign is the one given to $\mathcal{A}$ in **Game 1**. Namely, there is no problem for doing this since the two keys are uniformly distributed in the same key space.

Game 2. Here, $\mathcal{B}$ simulates the random oracle h as well as the random function $\mathcal{F}$ using two appropriate lists h-List and F-List. It will apply the following rules:

- To a query m_i, $\mathcal{B}$ picks h_i uniformly at random in $\mathcal{M}_{\mathrm{ex}}$ and adds the element (m_i, h_i) in h-List if m_i is not already in h-List. Otherwise, it simply looks in the h-List and answers the corresponding h-value.
- To handle the $\mathcal{F}$ and $\mathcal{F}^{-1}$ oracle queries, it proceeds in a similar way. To a query $(m_{\mathrm{un},i}, \sigma_{\mathrm{un},i})$, it picks f_i uniformly at random in $\mathcal{M}_{\mathrm{ex}}$ and adds $((m_{\mathrm{un},i}, \sigma_{\mathrm{un},i}), f_i)$ in F-List if $(m_{\mathrm{un},i}, \sigma_{\mathrm{un},i})$ is not already in F-List . Otherwise, $\mathcal{B}$ answers the corresponding f_i taken from F-List. Note that the simulation fails when collisions occur for some distinct f_i since $\mathcal{F}$ is a bijective function. It proceeds exactly in the same way for the $\mathcal{F}^{-1}$ queries by using the same list F-List.

Since h is a random oracle and $\mathcal{F}$ a random function oracle, we see that the simulation is perfect except when a collision on outputs of $\mathcal{F}$ resp. $\mathcal{F}^{-1}$ occurs.

Let CollF be the event that such a collision occurs in **Game 1** (equivalently in **Game 2**). Obviously, $\Pr[\mathsf{S}_1 \wedge \neg\mathsf{CollF}] = \Pr[\mathsf{S}_2 \wedge \neg\mathsf{CollF}]$, so we can apply the Shoup's lemma [26] and obtain

$$|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_1]| \leq \Pr[\mathsf{CollF}] \leq \frac{(q_{\mathcal{F}} + q_{\mathcal{F}}^*)^2}{|\mathcal{M}_{\mathrm{ex}}|}.$$

Game 3. This game is identical as **Game 2** except that $\mathcal{B}$ simulates the Sign oracle. Sign must query m_i^{s} to h. Let h_t be the answer. Sign must also run $\mathsf{UnForge}$. Let $(m'_{\mathrm{un},i}, \sigma'_{\mathrm{un},i})$ be the forged message-signature pair with respect to the Unsign scheme. It also runs the probabilistic algorithm $\mathsf{ExForge}$ which outputs a valid message-signature pair $(m_{\mathrm{ex},i}, \sigma_{\mathrm{ex},i})$ with respect to ExSign. Sign must also query $\mathcal{F}$ with $(m'_{\mathrm{un},i}, \sigma'_{\mathrm{un},i})$ and gets some f_s. Then, $\mathcal{B}$ simulates the value $f_s := \mathcal{F}(m'_{\mathrm{un},i}, \sigma'_{\mathrm{un},i})$ by setting $f_s := m_{\mathrm{ex},i} \odot (h_t)^{-1}$. Note that if $(m'_{\mathrm{un},i}, \sigma'_{\mathrm{un},i})$ or f_s is an element which lies already in $\mathsf{F\text{-}List}$ $\mathcal{B}$ has to abort the simulation. Namely, in the first case it could not choose the output value f_s while in the second case it might fail the simulation if f_s has a preimage which is not a valid message-signature pair in $\mathcal{M}_{\mathrm{un}} \times \Sigma_{\mathrm{un}}$. Since the collisions related to the outputs of $\mathcal{F}$ and $\mathcal{F}^{-1}$ (even those queried by ExSign) are already cancelled in **Game 2**, such bad events do not happen here. Hence, we notice that the simulation is perfect since $\mathsf{ExForge}$ outputs an $m_{\mathrm{ex},i}$ which is uniformly picked in $\mathcal{M}_{\mathrm{ex}}$. Note also that the distribution of $m'_{\mathrm{un},i}$ is uniform (assumed for $\mathsf{UnForge}$). Thus, for any h_t the distribution of f_s is uniform as well and the distribution of the pairs $(m_{\mathrm{ex},i}, \sigma_{\mathrm{ex},i})$ is the same as that from Sign. We have

$$\Pr[\mathsf{S}_3] = \Pr[\mathsf{S}_2].$$

Game 4. Here, we would like to obtain a game where the output forged message-signature pair $(m, \sigma) = (m, (m_{\mathrm{ex}}, \sigma_{\mathrm{ex}}))$ has the two following properties:

- m was queried to the random oracle h (necessarily not through Sign).
- $f := m_{\mathrm{ex}} \odot h(m)^{-1}$ is an output from a query made to the oracle $\mathcal{F}$ (maybe through Sign).

The first condition does not hold with a probability less than $1/|\mathcal{M}_{\mathrm{ex}}|$ since the attacker $\mathcal{A}$ could not do better than guessing the right $h(m)$. The second one does not hold if $\mathcal{A}$ guessed the right f (i.e., with probability up to $1/|\mathcal{M}_{\mathrm{ex}}|$) or if it queried f to $\mathcal{F}^{-1}$-oracle and obtained a valid signature pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$, i.e., with probability up to ν since UnSign is balanced. The probability that this condition does not hold is then less than $\max(1/|\mathcal{M}_{\mathrm{ex}}|, \nu)$ which is ν since $1/\nu < |\Sigma_{\mathrm{un}}| < |\mathcal{M}_{\mathrm{ex}}|$. Therefore,

$$|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_3]| \leq \frac{1}{|\mathcal{M}_{\mathrm{ex}}|} + \nu \leq 2\nu.$$

Game 5. $\mathcal{B}$ picks $j \in_U \{1, \ldots, q_h\}$, $\ell \in_U \{1, \ldots, q_{\mathcal{F}}\}$ at the beginning and it succeeds if m was the jth query to h and $m_{\mathrm{ex}} \odot h(m)^{-1}$ was the output from the ℓth query to $\mathcal{F}$. We have,

$$\Pr[\mathsf{S}_5] = \frac{1}{q_h \cdot q_{\mathcal{F}}} \Pr[\mathsf{S}_4].$$

Game 6. Here, $\mathcal{B}$ simulates the output h_j by setting $h_j := f_\ell^{-1} \odot m_{\text{chal}}$. This simulation is perfect because m_{chal} is an element uniformly picked at random and is unused so far. Thus,

$$\Pr[\mathsf{S}_6] = \Pr[\mathsf{S}_5].$$

Finally, we notice that $\mathcal{A}$ forged an ExSign signature to the message m_{chal} if it succeeds in the **Game 6** since $m = m_j$, $f = f_\ell$ and $m_{\text{ex}} = m_{\text{chal}}$ in this case. We then have $\Pr[\mathsf{Succ}^{\mathsf{uf-nma}}_{\mathsf{ExSign},\mathcal{B}}(k)] = \Pr[\mathsf{S}_6]$. Thus,

$$\Pr[\mathsf{Succ}^{\mathsf{uf-nma}}_{\mathsf{ExSign},\mathcal{B}}(k)] \geq \frac{1}{q_{\mathcal{F}} \cdot q_h} \left(\varepsilon - \frac{(q_{\mathcal{F}} + q^*_{\mathcal{F}})^2}{|\mathcal{M}_{\text{ex}}|} - 2\nu \right). \qquad \square$$

5.2 Invisibility to Lunchtime Chosen-Message Distinguisher

Theorem 5 (Invisibility). *Assume that h and C are fixed and that σ_{un} is uniformly distributed for any fixed key when m_{un} is uniformly distributed. For any invisibility distinguisher $\mathcal{D}$ under a lunchtime chosen-message attack against* Sign *with advantage $\varepsilon > 0$, there exists an invisibility distinguisher $\mathcal{UD}$ under a lunchtime known-message attack against* UnSign *with advantage $\varepsilon' \geq \varepsilon/2$ which uses one run of $\mathcal{D}$.*

Proof. First $\mathcal{UD}$ is fed with $\mathcal{K}_{\text{p}}^{\mathbf{C}}$ issued from $(\mathcal{K}_{\text{p}}^{\mathbf{C}}, \mathcal{K}_{\text{s}}^{\mathbf{C}}) \leftarrow \mathsf{Setup}^{\mathbf{C}}(1^k)$. Then, $\mathcal{UD}$ runs $(\mathcal{K}_{\text{p}}^{\mathbf{S}}, \mathcal{K}_{\text{s}}^{\mathbf{S}}) \leftarrow \mathsf{Setup}^{\mathbf{S}}(1^k)$ and transmits $\mathcal{K}_{\text{p}}^{\mathbf{C}}, \mathcal{K}_{\text{p}}^{\mathbf{S}}, \mathcal{K}_{\text{s}}^{\mathbf{S}}$ to $\mathcal{D}$. The answers of the oracle queries from $\mathcal{D}$ will be simulated by $\mathcal{UD}$. Since $\mathcal{D}$ has the signer secret key $\mathcal{K}_{\text{s}}^{\mathbf{S}}$, it does not need any access to a signing oracle. $\mathcal{UD}$ simulates the oracle queries to the confirmation and denial protocol as follows:

- To a message-signature pair $(m, (m_{\text{ex}}, \sigma_{\text{ex}}))$, $\mathcal{UD}$ checks first that $(m_{\text{ex}}, \sigma_{\text{ex}})$ is a valid pair with respect to ExSign. It retrieves the corresponding $(m_{\text{un}}, \sigma_{\text{un}})$ and forwards this query to the confirmation (or denial) protocol oracle with respect to UnSign.

At a time, $\mathcal{D}$ sends two messages $m_0, m_1 \in \mathcal{M}$ to $\mathcal{UD}$. $\mathcal{UD}$ receives from its challenger two messages $m_{\text{un}}^0, m_{\text{un}}^1 \in \mathcal{M}_{\text{un}}$ and a signature $\sigma_{\text{un}} \in \Sigma_{\text{un}}$ (The challenger flipped a coin $b \in_U \{0,1\}$ and set $\sigma_{\text{un}} \leftarrow \mathsf{UnSign}(m_{\text{un}}^b)$). Then, $\mathcal{UD}$ picks two random bits $b_1, b_2 \in_U \{0,1\}$, sets $m_{\text{ex}} = \mathcal{F}(m_{\text{un}}^{b_2}, \sigma_{\text{un}}) \odot h(m_{b_1})$, computes $\sigma_{\text{ex}} = \mathsf{ExSign}_{\mathcal{K}_{\text{s}}^{\mathbf{S}}}(m_{\text{ex}})$ and sends $\sigma = (m_{\text{ex}}, \sigma_{\text{ex}})$ to $\mathcal{D}$. Then, $\mathcal{D}$ answers a bit b'' to $\mathcal{UD}$. Finally, $\mathcal{UD}$ answers a bit $b' = b_1 \oplus b_2 \oplus b''$ (If $\mathcal{D}$ aborts, we pick a random b''.) to its challenger. It remains to compute the probability of success of $\mathcal{UD}$. To this end, we compute $\Pr[b' = b] = \Pr[b' = b \wedge b_2 = b] + \Pr[b' = b \wedge b_2 \neq b]$. We also have

$$\Pr[b' = b \wedge b_2 \neq b] = \Pr[b'' = b \oplus b_2 \oplus b_1 \wedge b_2 \neq b] = \Pr[b'' = \neg b_1 | b_2 \neq b] \cdot \frac{1}{2}.$$

When $b_2 \neq b$ then $(m_{\text{un}}^{b_2}, \sigma_{\text{un}})$ is uniformly distributed and independent from b_1, hence b'' is independent from b_1. Thus, $\Pr[b' = b \wedge b_2 \neq b] = 1/4$. Finally, since $\Pr[b' = b \wedge b_2 = b] = (1/2 + \varepsilon)\Pr[b_2 = b] = 1/2(1/2 + \varepsilon)$ we get $\Pr[b' = b] = 1/2 + \varepsilon/2$. $\square$

The scheme Sign does not satisfy the stronger adaptive invisibility notion defined in [3]. Namely, after having received the challenged signature σ, $\mathcal{D}$ could deduce the two pairs $(m_{\text{un}}^0, \sigma_{\text{un}}^0)$, $(m_{\text{un}}^1, \sigma_{\text{un}}^1)$ which would correspond to m_0 and m_1. Then, $\mathcal{D}$ generates a signature σ' on another message m' by using $(m_{\text{un}}^0, \sigma_{\text{un}}^0)$ and queries the pair (m', σ') to the confirmation and denial oracle. Depending on the answer, $\mathcal{D}$ deduces whether $(m_{\text{un}}^0, \sigma_{\text{un}}^0)$ is valid or not. From this, we see that $\mathcal{D}$ wins the invisibility game under an adaptive attack.

The fundamental problem relies on the fact that the attacker can always retrieve the corresponding pair $(m_{\text{un}}, \sigma_{\text{un}})$ (as any verifier) from a message-signature pair with respect to Sign. He can then sign a new message m' by reusing the pair $(m_{\text{un}}, \sigma_{\text{un}})$ and query the obtained pair to the Confirm or Deny oracle. Assuming that the verifier has to retrieve $(m_{\text{un}}, \sigma_{\text{un}})$, the only way to thwart such an attack is to make sure that the attacker cannot generate a new signature with another message m' with the same pair $(m_{\text{un}}, \sigma_{\text{un}})$. This seems to imply that $(m_{\text{un}}, \sigma_{\text{un}})$ has to depend on m. Moreover, the verifier should not be able to verify how $(m_{\text{un}}, \sigma_{\text{un}})$ was generated since it would trivially break the invisibility. This leads us to believe that the signer has to encrypt an element with the secret confirmer key such as in the scheme proposed in [3]. Obviously, the above discussion motivates the fact that we should strongly modify the generalized Chaum's scheme, in particular the confirmation (resp. denial) protocol cannot be achieved only with UnConfirm (resp. UnDeny).

5.3 Other Security Properties

The other security properties of our scheme are easier to prove, namely the completeness of the confirmation resp. denial protocol is straightforward. The other properties such as the soundness are inherited from the undeniable signature scheme. The non-transferability is also inherited. The non-coercibility is obtained if the signer deleted intermediate computations from UnForge. In this case, the invisibility of the undeniable signature scheme applies. Note that receipt-freeness is not guaranteed.

6 A Practical Example

Here, we propose a practical realization of the presented construction quite similar to that of Chaum [6]. First, we consider the Chaum's undeniable signature scheme [5] for UnSign. Let p be a prime integer of 1024 bits and g be a public generator of $\mathbb{Z}_p^*$. Then, $(\mathcal{K}_{\text{s}}^{\mathbf{C}}, \mathcal{K}_{\text{p}}^{\mathbf{C}}) = (c, g^c \bmod p) := (c, h)$ for a $c \in_U \mathbb{Z}_{p-1}^*$. We recall that Chaum's undeniable signature of a message $m_{\text{un}} \in \mathbb{Z}_p^*$ is $m_{\text{un}}^c \bmod p$. Hence, UnForge can be implemented by picking a random element $r \in \mathbb{Z}_{p-1}$ and outputting the pair $(m_{\text{un}}, \sigma_{\text{un}}) := (g^r \bmod p, h^r \bmod p)$. The random function $\mathcal{F}$ applied on $(m_{\text{un}}, \sigma_{\text{un}})$ can be implemented by computing an AES with a fixed key in a kind of CBC mode on $m_{\text{un}}||\sigma_{\text{un}}$ by $B(m_{\text{un}}||\sigma_{\text{un}}) = (x_0||\dots||x_{15})$ where $x_i \in \{0,1\}^{128}$ and $C(x_0||\dots||x_{15}) = (x_{16}||\dots||x_{31})$ with $x_i = \text{AES}(x_{i-16}) \oplus x_{i-1}$.

Note that we must choose p close enough to 2^{1024}. The hash function h can be instantiated with SHA-1 by $h(m) = \mathrm{trunc}_{2048}(\text{SHA-1}(1||m)||\ldots||\text{SHA-1}(13||m))$, where trunc_{2048} outputs the 2048 most significant bits of the input. The group operation $\odot$ can be replaced by the XOR operation $\oplus$ on the set $\{0,1\}^{2048}$. We finally take the plain DSA scheme for ExSign. Let q_1 be a prime integer close to 2^{2048}, a large prime number $q_2 = aq_1 + 1$ and a generator of $\mathbb{Z}_{q_2}^*$ whose a-th power is denoted as g_q. Then, $(\mathcal{K}_{\mathrm{s}}^{\mathbf{S}}, \mathcal{K}_{\mathrm{p}}^{\mathbf{S}}) = (x, g_q^x \bmod q_2)$ for $x \in_U \mathbb{Z}_{q_1}^*$. Then, $\sigma_{\mathrm{ex}} = (r,s)$, where $r = (g_q^k \bmod q_2) \bmod q_1$ and $s = \frac{m_{\mathrm{ex}}+xr}{k} \bmod q_1$ for a random $k \in_U \mathbb{Z}_{q_1}^*$.

7 On Feasibility Results based on Cryptographic Primitives

7.1 Discussion

This subsection provides a discussion on the relevance of the primitives used in the generalized Chaum's designated confirmer signature scheme. Namely, we would like to explain why this construction is possible although a previous result of Okamoto [19] seems at the first glance to provide strong evidence of its impossibility.

The study of relations between the cryptographic primitives always played a central role in cryptography. In particular, it allows to clarify the kind of primitives required to achieve the security of a given construction. Examples of well-known basic primitives are *one-way function*, *trapdoor one-way function*, or *trapdoor predicates* which were introduced by Goldwasser and Micali [10]. Here, we will focus on two classes of equivalent primitives, that of one-way functions and that of trapdoor predicates. These two classes contain respectively two major cryptographic primitives, namely the digital signatures resp. the public-key encryption. Rompel [24] proved that one-way functions are equivalent to signatures and Goldwasser and Micali [10] showed the equivalence between trapdoor predicates and public-key encryption. Since then, several cryptographic primitives have been shown to belong to one of these classes, e.g. undeniable signatures exist if and only if digital signatures exist [1].

Soon after their invention, designated confirmer signatures were proved to belong in the public-key encryption class [19]. This showed that despite of their similarities to undeniable signatures these two primitives are not equivalent. Separation between these two classes was proved by Impagliazzo et al. [13] in the black-box case, i.e., when the primitives are considered as black-box. This is quite relevant since almost all reductions considered in cryptography are black-box. Hence, this shows that the construction of a designated confirmer signature requires a primitive equivalent to the public-key encryption.

Our proposed construction seems only to be based on primitives belonging to the digital signatures class. Actually, this comes from an insufficient precise way to characterize cryptographic primitives. For instance, when we talk about a digital signature scheme, we mean a signature which is resistant to existential forgery under an adaptive chosen-message attack. Similarly an undeniable signature is meant to be implicitly secure in terms of existential forgery attacks and signatures invisibility. In this generalized Chaum's scheme, we have considered a special kind of undeniable signature which is existentially forgeable but remains invisible under a lunchtime known-message attack. In the next subsection, we prove that the existence of such a primitive indeed implies the existence of a public-key encryption semantically secure under a chosen-plaintext attack (IND-CPA). So we prove that undeniable signatures may belong to two different classes depending on the security properties we require. Paradoxically, although this kind of undeniable signature satisfies weaker security properties than usual, it belongs to a stronger class namely that of public-key encryption. Intuitively, this can be explained by the fact that it seems more difficult for an existentially forgeable undeniable signature to remain invisible than for an undeniable signature which is resistant to existential forgery attacks.

7.2 UnSign and Public-Key Encryption

We explain here how we can construct an IND-CPA public-key cryptosystem from the existentially forgeable undeniable signature scheme UnSign. We recall that UnSign is assumed to satisfy invisibility under a lunchtime known-message attack (this was required to prove that Sign is invisible under a lunchtime chosen-message attack). For the sake of simplicity, this cryptosystem will encrypt only one bit at a time. We denote the encryption scheme PKE. It is composed of three polynomial time algorithms which are the key generator KGen, the encryption algorithm Enc, and the decryption algorithm Dec. The scheme is inspired from [19].

KGen The key generator KGen generates a pair of key (pk, sk) by calling the key generator of UnSign. It computes $(\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{C}}) \leftarrow \mathsf{Setup}^{\mathbf{C}}(1^k)$ from the security parameter k and sets $(pk, sk) := (\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{C}})$.

Enc Let $b \in \{0, 1\}$ a bit to encrypt. If $b = 0$, we call the probabilistic algorithm UnForge to generate a valid pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}}) \leftarrow \mathsf{UnForge}(\mathcal{K}_{\mathrm{p}}^{\mathbf{C}})$. The pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ is set to be the ciphertext of b. If $b = 1$, we pick a pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}}) \in_U \mathcal{M}_{\mathrm{un}} \times \Sigma_{\mathrm{un}}$ uniformly at random. The pair $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ is the ciphertext of b in this case.

Dec Let $(m_{\mathrm{un}}, \sigma_{\mathrm{un}})$ be a ciphertext. Using the secret key $sk = \mathcal{K}_{\mathrm{s}}^{\mathbf{C}}$, it suffices to simulate UnConfirm or UnDeny to determine whether this pair is valid or not. If the pair is valid the decrypted ciphertext is 0, else it is 1.

We prove here that PKE is IND-CPA secure provided that UnSign is invisible under a lunchtime known-message attack. Assume the existence of an adversary $\mathcal{A}$ which wins in an IND-CPA game against PKE with a non-negligible advantage ε. Consider an adversary $\mathcal{B}$ which takes advantage of $\mathcal{A}$ in order to break the invisibility of UnSign under a lunchtime known-message attack.

At the beginning of the invisibility game, $\mathcal{B}$ receives a challenged pair of key $(\mathcal{K}_{\mathrm{p}}^{\mathbf{C}}, \mathcal{K}_{\mathrm{s}}^{\mathbf{C}})$ and playing the role of the challenger in the IND-CPA game forwards the same key pair to $\mathcal{A}$. After a given time, $\mathcal{A}$ will trivially send two bits $0, 1$

to $\mathcal{B}$. After a lunchtime, $\mathcal{B}$ will receive two challenged messages m^0_{un}, m^1_{un} with a signature σ_{un}. $\mathcal{B}$ sends the challenged pair $(m^0_{\text{un}}, \sigma_{\text{un}})$ to $\mathcal{A}$. Note that this challenge is perfectly simulated except when σ_{un} is a valid signature to both m^0_{un} and m^1_{un}. Such an event occurs with a probability ν. Otherwise, the probability for $(m^0_{\text{un}}, \sigma_{\text{un}})$ to be a valid message-signature pair is exactly $1/2$. Then, $\mathcal{A}$ answers a bit b. This bit b is also the answer of $\mathcal{B}$ to its challenger. Thus, the advantage ε' of $\mathcal{B}$ satisfies $\varepsilon' \geq \varepsilon - \nu$.

8 Conclusion

We revisited the designated confirmer signature scheme of Chaum and extended this one in a natural way in a generic scheme which transforms an undeniable signature scheme into a designated confirmer signature scheme. In the random oracle model, we proved that this construction is resistant against existential forgery under an adaptive chosen-message attack in which the attacker is the confirmer. It satisfies invisibility in the non-adaptive scenario in which the attacker is the signer. Our results trivially apply to the original Chaum scheme. Selective convertibility can also be included in this construction. As far as we know this construction is the only one which is based on a generic undeniable signature scheme and which is proven existentially unforgeable against an attacker having the confirmer's secret key. Finally, we proved that an existentially unforgeable undeniable signature which is invisible under a known-message attack scheme lies in the class of cryptographic primitives equivalent to the public-key encryption.

References

1. J. Boyar, D. Chaum, I. Damgård, and T. Pedersen, *Convertible Undeniable Signatures*, Advances in Cryptology - Crypto '90, LNCS **537**, pp. 189–205, Springer, 1991.
2. G. Brassard, D. Chaum, and C. Crépeau, *Minimum Disclosure Proofs of Knowledge*, Journal of Computer and System Sciences, vol. **37** (2), pp. 156-189, 1988.
3. J. Camenisch and M. Michels, *Confirmer Signature Schemes Secure against Adaptive Adversaries*, Advances in Cryptology - Eurocrypt '00, LNCS **1807**, pp. 243-258, Springer, 2000.
4. J. Camenisch and V. Shoup, *Practical Verifiable Encryption and Decryption of Discrete Logarithms*, Advances in Cryptology - Crypto '03, LNCS **2729**, pp. 126-144, Springer, 2003.
5. D. Chaum, *Zero-Knowledge Undeniable Signatures*, Advances in Cryptology - Eurocrypt '90, LNCS **473**, pp. 458-464, Springer, 1990.
6. D. Chaum, *Designated Confirmer Signatures*, Advances in Cryptology - Eurocrypt '94, LNCS **950**, pp. 86-91, Springer, 1995.
7. D. Chaum and H. van Antwerpen, *Undeniable Signatures*, Advances in Cryptology - Crypto '89, LNCS **435**, pp. 212-217, Springer, 1989.
8. S. Galbraith and W. Mao, *Invisibility and Anonymity of Undeniable and Confirmer Signatures*, CT-RSA 2003, LNCS **2612**, pp. 80-97, Springer, 2003.

9. R. Gennaro, T. Rabin, and H. Krawczyk, *RSA-Based Undeniable Signatures*, Journal of Cryptology, vol. **13** (4), pp. 397-416, Springer, 2000.
10. S. Goldwasser and S. Micali, *Probabilistic encryption*, Journal of Computer and System Sciences, vol. **28** (2), pp. 270-299, 1984.
11. S. Goldwasser, S. Micali, and R. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM Journal on Computing, vol. **17** (2), pp. 281-308, 1988.
12. S. Goldwasser and E. Waisbard, *Transformation of Digital Signature Schemes into Designated Confirmer Signatures Schemes*, TCC '04, LNCS **2951**, pp. 77-100, Springer, 2004.
13. R. Impagliazzo and S. Rudich, *Limits on the Provable Consequences of One-way Permutations*, 21st Annual ACM Symposium on Theory of Computing, pp. 44-61, ACM Press, 1989.
14. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated Verifier Proofs and Their Applications*, Advances in Cryptology - Eurocrypt '96, LNCS **1070**, pp. 143-154, Springer, 1996.
15. B. Libert and J.-J. Quisquater, *Identity Based Undeniable Signatures*, CT-RSA '04, LNCS **2964**, pp. 112-125, Springer, 2004.
16. M. Michels and M. Stadler, *Generic Constructions for Secure and Efficient Confirmer Signatures Schemes*, Advances in Cryptology - Eurocrypt '98, LNCS **1403**, pp. 406-421, Springer, 1998.
17. J. Monnerat and S. Vaudenay, *Undeniable Signatures Based on Characters*, PKC '04, LNCS **2947**, pp. 69-85, Springer, 2004.
18. J. Monnerat and S. Vaudenay, *Generic Homomorphic Undeniable Signatures*, Advances in Cryptology - Asiacrypt '04, LNCS **3329**, pp. 354-371, Springer, 2004.
19. T. Okamoto, *Designated Confirmer Signatures and Public-key Encryption are Equivalent*, Advances in Cryptology - Crypto '94, LNCS **839**, pp. 61-74, Springer, 1994.
20. T. Okamoto and D. Pointcheval, *The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes*, PKC '01, LNCS **1992**, pp. 104-118, Springer, 2001.
21. D. H. Phan and D. Pointcheval, *Chosen-Ciphertext Security without Redundancy*, Advances in Cryptology - Asiacrypt '03, LNCS **2894**, pp. 1-18, Springer, 2003.
22. D. Pointcheval and J. Stern, *Security Arguments for Digital Signatures and Blind Signatures*, Journal of Cryptology, vol. **13** (3), pp. 361-396, 2000.
23. R. Rivest, A. Shamir, and A. Tauman, *How to Leak a Secret*, Advances in Cryptology - Asiacrypt '01, LNCS **2248**, pp. 552-565, Springer, 2001.
24. J. Rompel, *One-Way Functions are Necessary and Sufficient for Secure Signatures*, 22nd Annual ACM Symposium on Theory of Computing, pp. 387-394, ACM Press, 1990.
25. K. Sakurai and S. Miyazaki, *An Anonymous Electronic Bidding Protocol Based on a New Convertible Group Signature Scheme*, ACISP '00, LNCS **1841**, pp. 385-399, Springer, 2000.
26. V. Shoup, *Sequences of Games: a Tool for Taming Complexity in Security Proofs*, Cryptology ePrint Archive, Report 2004/332, `http://eprint.iacr.org/`, 2004.