Short Undeniable Signatures Without Random Oracles: The Missing Link

Fabien Laguillaumie[1] and Damien Vergnaud[2]

[1] Projet TANC - INRIA Futurs, Laboratoire d'informatique (LIX)
École polytechnique, 91128 Palaiseau cedex, France
laguillaumie@lix.polytechnique.fr

[2] Laboratoire de Mathématiques Nicolas Oresme,
Université de Caen, Campus II,
B.P. 5186, 14032 Caen Cedex, France
vergnaud@math.unicaen.fr

Abstract. We introduce a new undeniable signature scheme which is existentially unforgeable and anonymous under chosen message attacks in the standard model. The scheme is an embedding of Boneh and Boyen's recent short signature scheme in a group where the *decisional Diffie-Hellman problem* is assumed to be difficult. The anonymity of our scheme relies on a decisional variant of the *strong Diffie-Hellman assumption*, while its unforgeability relies on the strong Diffie-Hellman assumption.

1 Introduction

We design new undeniable signatures. Our approach is both practical and theoretical: we build a very efficient protocol with short signatures and analyze its security in the complexity theory setting (*i.e* with reductionist proofs). The security (in the sense of unforgeability and anonymity) relies on *strong Diffie-Hellman assumptions* in the standard model. It is worth noting that the new mechanism is the first efficient scheme proven secure without any random oracles.

Related work. The *self-authenticating* property of digital signatures can be suitable for many applications such as dissemination of public-key certificates or official announcements, but seems undesirable in personally or commercially sensitive applications. Therefore it may be preferable to put some restrictions on this property to prevent misuse of signatures. Undeniable signatures were introduced in 1989 by Chaum and van Antwerpen [6] to limit the self-authenticating property of digital signatures. In this setting, one has to interact with the signer in order to be convinced of the validity of a given signature. The security of the seminal protocol relies on the discrete logarithm problem, but suffers from the fact that the interactive protocols were not zero-knowledge. In 1990, Chaum [5] improved the initial proposal by providing a zero-knowledge version. The security of Chaum and van Antwerpen's undeniable signatures was eventually proven by Okamoto and Pointcheval in 2001 [20], using the so-called *gap-problems*. In

S. Maitra et al. (Eds.): INDOCRYPT 2005, LNCS 3797, pp. 283–296, 2005.

[21], Ogata, Kurosawa, and Heng showed that the security can in fact be proven under a classical computational assumption. This concept has been investigated for years, and many proposals appear in the literature. In 1991, the concept has been refined by giving the possibility to transform an undeniable signature into a *self-authenticating* signature. These *convertible undeniable signatures*, proposed in [3] by Boyar, Chaum, Damgård and Pedersen, provide individual and universal conversions of the signatures. They were broken and repaired in 1996 by Michels, Petersen and Horster [17]. Several schemes were subsequently proposed, based on well-known signatures [10, 9, 8]. Recently, an identity-based undeniable signature scheme built on bilinear maps was proposed by Libert and Quisquater [16]. Monnerat and Vaudenay [18, 19] proposed short undeniable signatures based on characters (without the conversion property). Finally, we extended in [15] the concept of convertible undeniable signatures by giving the signer the ability to convert signatures pertaining to a specific time period.

Our contributions. In groups where there exists an oracle for the decisional Diffie-Hellman problem, Chaum and van Antwerpen's undeniable signatures become self-authenticating. They were revisited by Boneh, Lynn and Shacham (BLS) in 2001 [2] and considered on groups where there exists an admissible bilinear map. An elegant variant of these signatures, still pairing-based, was introduced in 2004 by Boneh and Boyen [1] (and also by Zhang, Safavi-Naini and Susilo [23]). Its unforgeability was proven in the standard model. Contrary to the latter approach, in this article, we remove the bilinear map from Boneh-Boyen signatures to obtain the first efficient undeniable signature scheme without random oracle.

In [12], Goldwasser and Waisbard proposed *designated confirmer signatures* without random oracles. Their techniques could be extended to construct secure undeniable signatures. Indeed, this transformation is straightforward, since their general construction remains secure if the designated confirmer is the signer himself. Goldwasser and Waisbard do not give efficient disavowal protocols for their instanciations. They argued for designated confirmer signatures there is no need for such a protocol. However, to get an efficient complete undeniable signature protocol, there are a number of non-trivial details that would need to be worked out. The resulting schemes will be far less efficient than our proposal.

Our undeniable signatures are to Chaum and van Antwerpen undeniable signatures what Boneh-Boyen's signatures are to BLS. In the dual way, the new scheme is to Boneh-Boyen's scheme what Chaum and van Antwerpen's construction is to BLS.

The security of previous proposals of undeniable signatures is carried in the random oracle model and therefore is only heuristic. Like Boneh-Boyen's scheme, the security of our protocol does not rely on any idealized primitive but needs stronger computational assumptions. As pointed out in [14], the confirming and denying protocols are important elements in the security analysis of an undeniable signature scheme. The main difficulties to study our scheme in the standard security model, arise in the simulation of the *interactive* confirmation and denying protocols in the reductionist proof. The present paper provides the first security analysis for undeniable signatures in this interactive setting.

2 Preliminaries

2.1 Undeniable Signatures

Definition 1 (Undeniable Signatures). *Given an integer k, an* undeniable signature scheme *US with security parameter k is defined by the following:*

- *a* **common parameter generation algorithm** *US.Setup: it is a probabilistic algorithm which takes as input k. The outputs are* the public parameters*;*
- *a* **key generation algorithm for the signers** *US.SKeyGen: it is a probabilistic algorithm which takes as input the public parameters and outputs a pair of keys (pk, sk);*
- *a* **key generation algorithm for a verifiers** *US.VKeyGen: it is a probabilistic algorithm which takes as input the public parameters and outputs a pair of keys (pk, sk);*
- *a* **key registration protocol** *US.Register is a protocol between a "key registration authority" (KRA) and a verifier with common input the public parameters. At the end, the KRA outputs a pair (pk, notif) where pk is the verifier's public key and $\mathsf{notif} \in \{0,1\}^*$ is a key registration authorization decision.*
- *a* **signing algorithm** *US.Sign: it is a probabilistic algorithm which takes as input a message m, a secret key sk and the public parameters. The output σ is* an undeniable signature on *m;*
- **confirming/denying protocols** *US.{Confirm.Deny}: they are protocols which take as input a message m, a putative undeniable signature σ on m, a pair of keys (pk, sk) and the public parameters. The output is a (possibly non-interactive) non-transferable proof that σ is actually a valid/invalid undeniable signature on m, with respect to the key pk;*

and must satisfy the following properties (formally defined below):

1. *correctness and soundness: the confirming and denying protocols and the verifying algorithms are complete and sound, where completeness means that valid (invalid) signatures can always be proved valid (invalid), and soundness means that no valid (invalid) signature can be proved invalid (valid);*
2. *unforgeability: given a public key, it is computationally infeasible, without the knowledge of the corresponding secret key to produce an undeniable signature that is accepted by the verification algorithm or by the confirming protocol;*
3. *anonymity: given a message m and an undeniable signature σ on m, it is computationally infeasible to find which secret key was used to generate σ;*
4. *non-transferability: a verifier participating in an execution of the confirming/denying protocols does not obtain information that could be used to convince a third party about the validity/invalidty of a signature.*

Remark 1. The aim of the protocol Register is to force the verifiers to "know" the secret key corresponding to their public key, in order to enforce the non-transferability property. We assume for simplicity that the verifier just reveals his key pair (pk, sk) and the key registration authority authorizes it if and only if $(pk, sk) \in \mathsf{US.VKeyGen}(\mathtt{params})$.[1]

[1] This can always be done, since we can assume that the secret key contains the randomness input used to generate it.

2.2 Security Model

Anonymity. The notion of *anonymity* under a chosen message attack ($\mathsf{Ano\text{-}CMA}$) was precisely defined by Galbraith and Mao in [8]. An $\mathsf{Ano\text{-}CMA}$-adversary $\mathcal{A}$ runs in two stages. In the `find` stage, it takes as input two public keys pk_0 and pk_1 and outputs a message $m^\star$ (and some state information s). In the `guess` stage it gets a challenge undeniable signature $\sigma^\star$ formed by signing the message $m^\star$ at random under one of the two keys, and must say which key was chosen. In both stages, the adversary has access to the signing oracles Σ_0, Σ_1, to the confirming oracles $\Upsilon_{C,0}$ and $\Upsilon_{C,1}$ (with registered verifying keys) and to the denying oracles $\Upsilon_{D,0}$ and $\Upsilon_{D,1}$ (with registered verifying keys). The only restriction is that he cannot query the couple $(m^\star, \sigma^\star)$ on the confirming/denying oracles.

Definition 2 (Anonymity). *Let US be an undeniable signature scheme and let $\mathcal{A}$ be an $\mathsf{Ano\text{-}CMA}$-adversary against US. We consider the following two random experiments, for $r \in \{0,1\}$, where k is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{ano\text{-}cma}-r}_{\mathsf{US},\mathcal{A}}(k)}$
$\text{params} \overset{R}{\longleftarrow} \mathsf{US.Setup}(k)$
$(pk_0, sk_0) \overset{R}{\longleftarrow} \mathsf{US.KeyGen}(\text{params})$
$(pk_1, sk_1) \overset{R}{\longleftarrow} \mathsf{US.KeyGen}(\text{params})$
$(m^\star, s) \leftarrow \mathcal{A}^{\Sigma_0,\Sigma_1,\Upsilon_{C,0},\Upsilon_{C,1},\Upsilon_{D,0},\Upsilon_{D,1}}(\texttt{find}, \text{params}, pk_0, pk_1)$
$\sigma^\star \leftarrow \mathsf{US.Sign}(\text{params}, m, sk_r)$
$d \leftarrow \mathcal{A}^{\Sigma_0,\Sigma_1,\Upsilon_{C,0},\Upsilon_{C,1},\Upsilon_{D,0},\Upsilon_{D,1}}(\texttt{guess}, \text{params}, pk_0, pk_1, m^\star, \sigma^\star)$
Return $d$
```

We define the advantage *of $\mathcal{A}$* via*:*

$$\mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{US},\mathcal{A}}(k) = \left|\Pr\left[\mathbf{Exp}^{\mathsf{ano-cma}-1}_{\mathsf{US},\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{ano-cma}-0}_{\mathsf{US},\mathcal{A}}(k) = 1\right]\right|.$$

Given $(k, \tau) \in \mathbb{N}^2$ and $\varepsilon \in [0,1]$, the scheme US is said to be (k, τ, ε)-$\mathsf{Ano\text{-}CMA}$ secure, if no $\mathsf{Ano\text{-}CMA}$-adversary $\mathcal{A}$ running in time τ has an advantage $\mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{US},\mathcal{A}}(k) \geq \varepsilon$.

Security against existential forgery under chosen message attack. Security for digital signatures was defined by Goldwasser, Micali and Rivest [11] as *existential forgery against adaptive chosen message attacks* ($\mathsf{EF\text{-}CMA}$). For undeniable signatures, unforgeability security is defined along the same lines, with the notable difference that, while mounting a chosen-message attack, we suppose that the adversary is allowed to query a confirming (*resp.* a denying) oracle Υ_C (*resp.* Υ_D) on any couple message/signature of his choice, in addition to the classical access to the signing oracle Σ. As usual, in the adversary answer, there is the natural restriction that in the returned couple message/signature $(m^\star, \sigma^\star)$, the message $m^\star$ has not been queried to the oracle Σ.

Definition 3 (Unforgeability). *Let US be an undeniable signature scheme and let $\mathcal{A}$ be an $\mathsf{EF\text{-}CMA}$-adversary against US. We consider the following random experiment, where k is a security parameter:*

$$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{ef-cma}}_{\mathit{US},\mathcal{A}}(k)}$$

$$\begin{array}{l}
\text{params} \overset{R}{\longleftarrow} \mathit{US.Setup}(k) \\
(pk, sk) \overset{R}{\longleftarrow} \mathit{US.KeyGen}(\text{params}) \\
(m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\Sigma, \Upsilon_C, \Upsilon_D}(\text{params}, pk) \\
\textit{Return } 1 \textit{ if } \sigma^\star \textit{ is a valid signature on } m^\star \\
\qquad\quad\; 0 \textit{ otherwise}
\end{array}$$

We define the success *of* $\mathcal{A}$ via $\mathbf{Succ}^{\textit{ef-cma}}_{\mathit{US},\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\textit{ef-cma}}_{\mathit{US},\mathcal{A}}(k) = \mathtt{valid}\right]$.

Given $(k, \tau) \in \mathbb{N}^2$ *and* $\varepsilon \in [0, 1]$*, the scheme* US *is said to be* (k, τ, ε)*-EF-CMA secure, if no EF-CMA-adversary* $\mathcal{A}$ *running in time* τ *has a success* $\mathbf{Succ}^{\textit{ef-cma}}_{\mathit{US},\mathcal{A}}(k) \geq \varepsilon$*.*

2.3 Proof of Knowledge

We cannot replace the zero-knowledge interactive proofs by non-interactive non-transferable proofs, to obtain the security results in the standard model. As far as we know, all these non-interactive proofs are either highly inefficient or obtained by applying the Fiat-Shamir heuristic to interactive designated-verifier proofs, and therefore their security relies on the random oracle paradigm.

Let $\mathbb{G}$ be a group. To confirm or deny that a bit string is a signature in the new undeniable signature scheme, it is necessary to prove that a given quadruple $(L, M, N, O) \in \mathbb{G}^4$ is a Diffie-Hellman quadruple (or not). To face *blackmailing* or *mafia* attacks against our undeniable signatures, we use interactive designated verifier proofs, as introduced in [13] by Jakobsson, Sako, and Impagliazzo, in Chaum's proofs of equality (*cf.* Fig. 1) and inequality (*cf.* Fig. 2) of discrete logarithm of [5]. The idea is to replace the generic commitment scheme by a trapdoor commitment, as defined in [4], and using classical techniques, the proofs are readily seen to be complete, sound, and above all non-transferable. The

$$\begin{array}{rcl}
\text{Prover} & & \text{Verifier} \\
\hline
 & & (u, v) \in_R [\![1, q-1]\!]^2 \\
 & \xleftarrow{\qquad\quad C \qquad\quad} & C = uL + vN \\
r \in_R [\![1, q-1]\!] & & \\
R_1 = C + rP_V & & \\
R_2 = xR_1 & \xrightarrow{\qquad R_1, R_2 \qquad} & \\
 & \xleftarrow{\qquad\quad u, v \qquad\quad} & \\
C \overset{?}{=} uL + vN & \xrightarrow{\qquad\quad r \qquad\quad} & \\
 & & R_1 \overset{?}{=} C + rP_V \\
 & & R_2 \overset{?}{=} (u + yr)M + vO \\
\hline
\end{array}$$

Fig. 1. ZKIP protocol to prove that $x = \log_L(M) = \log_N(O)$

Prover		Verifier
		$s \in_R [\![0, \lambda]\!]$ and $u \in_R [\![1, q-1]\!]$
	$\xleftarrow{\quad C, C' \quad}$	$C = uL + sN$ and $C' = uM + sO$
find s' such that $(xC - C') = s'(xN - O)$		
$r \in_R [\![1, q-1]\!]$		
$C'' = s'L + rP_{\mathcal{V}}$	$\xrightarrow{\quad C'' \quad}$	
	$\xleftarrow{\quad u \quad}$	
$C \overset{?}{=} uL + s'N$	$\xrightarrow{\quad (s', r) \quad}$	$C'' \overset{?}{=} s'L + rP_{\mathcal{V}}$ and $s \overset{?}{=} s'$

Fig. 2. ZKIP protocol to prove that $x = \log_L(M) \neq \log_N(O)$

Table 1. Some values of λ and computational workload in the proof of inequality

security		λ	$\lambda = 31$	$\lambda = 1023$
2^{30}	#iterations	$30/\log_2(\lambda+1)$	6	3
	#exp.	$\lambda/2 \times$ #iterations	93	1534.5
2^{80}	#iterations	$80/\log_2(\lambda+1)$	16	8
	#exp.	$\lambda/2 \times$ #iterations	248	4092

protocols, involve a point $P_{\mathcal{V}} = yL$ where y is the secret key of the verifier, and the prover must be convinced that $P_{\mathcal{V}}$ is well-formed (in the undeniable signature scheme, the registration protocol is used to force the users to know the secret-key corresponding to their public key).

In both protocols, the prover is given (L, M, N, O), and he knows $x = \log_L(M)$. As argued in [5], in the proof of inequality, the prover can cheat with probability $(\lambda + 1)^{-1}$. This leads to the table 1 with examples for suitable λ together with the round and computational complexities.

2.4 Underlying Problems

The security of asymmetric cryptographic tools relies on assumptions about the hardness of certain algorithmic problems. Throughout the paper $\mathbb{G}$ denotes an additive group of prime order q (*e.g.* the group of points of an elliptic curve over a finite field, a subgroup of the multiplicative group of a finite field). Our scheme relies on the difficulty of the algorithmic problems described below in $\mathbb{G}$ but on no other special property. Therefore, we choose not to pin down a specific group and to describe the protocol in a generic way:

Definition 4. *A* prime-order-group-generator *(POGG) is a probabilistic algorithm that takes a security parameter k as input and outputs a pair $(q, \mathbb{G})$ where q is a prime with $2^k < q < 2^{k+1}$, $\mathbb{G}$ is a group of order q.*

Let P be a generator of $\mathbb{G}$. In [1], Boneh and Boyen introduced a new computational problem in a bilinear context. For our purpose, we consider this problem in the classical discrete log setting, *i.e* without bilinear map.

ℓ-Computational Strong Diffie-Hellman (ℓ-CSDH): let x be in $[\![1, q-1]\!]$. Given an integer $\ell \in \mathbb{N}$ and $(P, xP, x^2P, \dots, x^\ell P) \in \mathbb{G}^{\ell+1}$, compute a pair $\big((x+h)^{-1}P, h\big)$ in $\mathbb{G} \times [\![1, q-1]\!]$ for some $h \in [\![1, q-1]\!]$.

The invisibility of our protocol relies on the decisional variant of this problem:

ℓ-Decisional Strong Diffie-Hellman (ℓ-DSDH): let x be in $[\![1, q-1]\!]$. Given an integer $\ell \in \mathbb{N}$ and $(P, xP, x^2P, \dots, x^\ell P) \in \mathbb{G}^{\ell+1}$, and a pair (Q, h) in $\mathbb{G} \times [\![1, q-1]\!]$ for some $h \in [\![1, q-1]\!]$, decide whether $Q = (x+h)^{-1}P$.

In [20], Okamoto and Pointcheval proved the security of the FDH variant of Chaum and van Antwerpen's undeniable signatures by introducing a new class of computational problems, called *gap problems*. In [21], Ogata, Kurosawa and Heng proved that the unforgeability of the protocol is equivalent to the classical CDH problem. In the context of undeniable signatures, the confirming and denying protocols can be executed on any message/putative signature chosen by the adversary. To take into account this kind of oracle access, we have to introduced a *gap-variant* of the Strong Diffie-Hellman problem if we do not want to lose a large factor in the unforgeability security reduction to CSDH.

ℓ-Gap Strong Diffie-Hellman (ℓ-GSDH): let x be in $[\![1, q-1]\!]$. Given an integer $\ell \in \mathbb{N}$ and $(P, xP, x^2P, \dots, x^\ell P) \in \mathbb{G}^{\ell+1}$, compute a pair $\big((x+h)^{-1}P, h\big)$ in $\mathbb{G} \times [\![1, q-1]\!]$ for some $h \in [\![1, q-1]\!]$, with the help of a ℓ-DSDH oracle.

3 Short Undeniable Signatures Without Random Oracles

3.1 The New Scheme

In this section, we describe our new undeniable signature scheme, parameterized by a prime-order-group-generator Gen. Note, that as mentioned above, for this basic version of the scheme, we use the direct key registration.

Common parameter generation algorithm $\mathsf{US.Setup}$: on input a security parameter k, the algorithm $\mathsf{Gen}(k)$ is run to produce a pair $(q, \mathbb{G})$. An element P is picked at random in $\mathbb{G} \setminus \{0_\mathbb{G}\}$ and the public parameters are $(q, \mathbb{G}, P)$.

Key generation algorithm for the signers $\mathsf{US.SKeyGen}$: Alice picks at random her secret key $(a_1, a_2) \in [\![1, q-1]\!]^2$ and sets (P_1, P_2) as her public key, with $P_1 = a_1P$ and $P_2 = a_2P$.

Key generation algorithm for the verifiers $\mathsf{US.VKeyGen}$: Bob picks at random his secret key $b \in [\![1, q-1]\!]$ and sets $P_B = bP$ as his public key.

Signing algorithm $\mathsf{US.Sign}$: to sign a message $m \in [\![1, q-1]\!]$, Alice picks at random $r \in [\![1, q-1]\!]$ and sets $S = (a_1+m+ra_2)^{-1}P$. The signature is $\sigma = (S, r)$.

Confirming/Denying protocol $\mathsf{US.\{Confirm, Deny\}}$: given a message m and a putative signature $\sigma = (S, r)$ on m, Alice proves to Bob (who has published a registered valid public key P_B) that $\log_S(P - mS) = \log_P(P_1 + rP_2)$ or not, using the protocols described in section 2.3. For the sake of simplicity, we suppose that the signer do not interleave several instances asynchronously nor concurrently.

Remark 2. The notion of on-line/off-line signatures was introduced by Even, Goldreich and Micali [7]. The idea is to generate signatures in two phases. The first one is performed off-line (*i.e.* before the message to be signed is given) and the second phase is performed on-line (once the message to be signed is known). On-line/Off-line signatures are useful since in many applications the signer has a very limited response time once the message is presented but he can carry out costly computations between consecutive signing requests.

Using the *sign and switch* paradigm, we can convert our undeniable signature scheme into a highly efficient on-line/off-line scheme. The signer computes off-line $S = (a_1 + t)^{-1}P$ for a random $t \in [\![1, q-1]\!]$. Once he is given the message m, the signature is completed with $r = a_2^{-1}(t - m)$ where $a_2^{-1} \mod q$ can also be precomputed. The on-line signature completion procedure then amounts to computing a hash value, a substraction and a multiplication modulo q.

Remark 3. One may require, of course, unforgeability and anonymity of the undeniable signatures, even against the key registration authority. To achieve this, one can replace the direct key registration protocol by a zero-knowledge proof of knowledge of the verifier's secret key (using for instance the Schnorr proof of knowledge of discrete logarithms [22]). The unforgeability and the anonymity of the scheme, can still be proved in the standard security model (by using rewinding techniques). Details on the security arguments will be given in the full version of the paper.

3.2 Security Results

Anonymity. For any Ano-CMA adversary $\mathcal{A}$, we denote by $\mathsf{Bad}_{\mathcal{A}}$ the event that $\mathcal{A}$ queries a valid signature to the confirming/denying oracle, which has not been obtained from the signing oracle.

Proposition 1. *Let Gen be a POGG and let US be the associated undeniable signature scheme. For any Ano-CMA adversary $\mathcal{A}$ against US, with security parameter k, which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{ano-cma}}_{\mathit{US},\mathcal{A}}(k)$, running time τ, making q_Σ queries to the signing oracle, q_C to the confirming oracle, q_D to the denying oracle and registers up to $q_\mathcal{R}$ keys, there exists an adversary $\mathcal{B}$ against $(q_\Sigma + 1)$-DSDH of advantage $\varepsilon' = \mathbf{Adv}^{(\mathsf{q}_\Sigma+1)-\mathsf{dsdh}}_{\mathit{Gen},\mathcal{B}}(k)$ and running time τ', such that $\varepsilon' \geq \varepsilon/2 - (q_\Sigma + 2)2^{-k} - \Pr[\mathit{Bad}]$ and $\tau' \leq q_C\tau + q_\Sigma(q_\Sigma T_\mathbb{G} + O(1))$, where $T_\mathbb{G}$ denotes the time complexity to perform a scalar multiplication in $\mathbb{G}$.*

Proof. Let k be a security parameter, $(q, \mathbb{G})$ be a couple generated by Gen. We consider a random instance of ℓ-DSDH denoted by $(P, xP, x^2P, \ldots, x^\ell P, Q, h)$ and we may assume that $q_\Sigma = \ell + 1$. We denote by A_i the point x^iP, for all $i \in [\![0, q_\Sigma - 1]\!]$. We construct a simulation which solves this instance.

Game_0 We consider an Ano-CMA-adversary $\mathcal{A}$ with advantage $\mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{US},\mathcal{A}}(k)$, within time τ. The key generation algorithm is run twice to produces two pairs of keys (pk_0, sk_0) and (pk_1, sk_1). In his first stage, the adversary $\mathcal{A}$ is fed with pk_0 and pk_1, and, querying the signing oracles Σ_0 and Σ_1, and the confirming/denying oracles $\Upsilon_{C,0}$, $\Upsilon_{C,1}$, $\Upsilon_{D,0}$ and $\Upsilon_{D,1}$, outputs a message $m^\star$. A challenger picks $b^\star \in \{0,1\}$ at random and queries the signing oracle $\Sigma_{b^\star}$ on $m^\star$ and sets the answer as $\sigma^\star$. In its second stage the attacker is given $\sigma^\star$ and has once more a permanent access to the oracles, with the natural restriction not to query the challenge signature on the confirming/denying oracles. We denote by q_Σ the number of queries to the signing oracle and q_C to the confirming oracle and q_D to the denying oracle.

In any game Game_i, we denote by $\mathsf{Guess_i}$ the event $b^\star = b$. By definition, $|2\Pr[\mathsf{Guess}_0] - 1| = \mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{USBM},\mathcal{A}}(k)$.

Game_1 First, $\mathcal{B}$, picks at random $(\alpha, a_0, a_1) \in [\![1, q-1]\!]^3$, initializes a counter c to the value 1 and two lists Σ-$\mathsf{List} = \{\}$ and $\overline{\Sigma}$-$\mathsf{List} = \{\}$. $\mathcal{B}$ prepares q_Σ random elements $h_i \in [\![1, q-1]\!]$, for $i \in [\![1, q_\Sigma]\!]$. If $h \in \{h_1, \ldots, h_{q_\Sigma}\}$ or $(\alpha h \mod q) \in \{h_1, \ldots, h_{q_\Sigma}\}$, $\mathcal{B}$ aborts: this happens with probability at most $q_\Sigma 2^{1-k}$. $\mathcal{B}$ computes the following polynomial:

$$f(y) = \prod_{i=1}^{q_\Sigma}(y + h_i) = \sum_{i=0}^{q_\Sigma} \alpha_i y^i \in \mathbb{F}_q[y], \text{ and the points } P = \sum_{i=0}^{q_\Sigma} \alpha_i A_i = f(x)P,$$

$$P_0 = \sum_{i=1}^{q_\Sigma+1} \alpha_{i-1} A_i = xf(x)P = xP' \text{ and } P_1 = \alpha P_0 = \alpha x P'. \text{ Finally}$$

$\mathcal{B}$ sets $\mathtt{params} = (q, \mathbb{G}, P')$, $pk_0 = (P_0, a_0 P')$, $pk_1 = (P_1, a_1 P')$ and feeds $\mathcal{A}$ with pk_0 and pk_1. The distribution of (pk_0, pk_1) is unchanged since $(A_0, \ldots, A_\ell, Q, h)$ is a random instance of the ℓ-DSDH problem and (α, a_0, a_1) is picked at random. Therefore we have $\Pr[\mathsf{Guess}_1] = \Pr[\mathsf{Guess}_0]$.

Game_2 From this game, $\mathcal{B}$ performs a specific stage that allows it to retrieve each verifier secret key y. If the direct key registration is used, then this is straightforward, otherwise, $\mathcal{B}$ might have to replay the simulation once with the same random tape such that monitoring the re-registration of $P_\mathcal{A}$ by $\mathcal{A}$ allows to extract y. In our case, this simulation is obviously perfect, therefore we obtain $\Pr[\mathsf{Guess}_2] = \Pr[\mathsf{Guess}_1]$.

Game_3 Now $\mathcal{B}$ simulates the signing oracles. It initializes a counter to $c = 1$, and for each new request $m \in \{0,1\}^*$, $\mathcal{B}$ constructs this polynomial of $\mathbb{F}_q[y]$: $f_c(y) = \dfrac{f(y)}{y + h_c} = \prod_{\substack{j=1 \\ j \neq c}}^{q_\Sigma}(y + h_j) = \sum_{j=0}^{q_\Sigma - 1} \beta_j^{(c)} y^j$ and then computes

$$S_c = \frac{1}{\alpha^b} \sum_{j=0}^{q_\Sigma - 1} \beta_j^{(c)} A_j = \frac{1}{\alpha^b (x + h_c)} P'$$ where Σ_b, with $b \in \{0,1\}$ is the oracle queried. Then $\mathcal{B}$ sets $r_c = (h_c \alpha^b - m_c) a_b^{-1}$. If $r_c = 0$, then $\mathcal{B}$ aborts, else it outputs (S_c, r_c) as a valid signature on m_c for the public key pk_b and adds $(m_c, (S_c, r_c), b)$ in the Σ-List. $\mathcal{B}$ increments the counter. During its whole execution, $\mathcal{B}$ reports failure in the signing simulation with prob-

ability at most $q_\Sigma 2^{-k}$. This game perfectly simulates the signing oracle if it does not abort. Therefore $|\Pr[\mathsf{Guess}_3] - \Pr[\mathsf{Guess}_2]| \leq q_\Sigma 2^{-k}$.

Game_4 When the adversary queries the confirming oracle $\Upsilon_{C,b}$ with $b \in \{0,1\}$ on a putative signature σ on m, $\mathcal{B}$ checks whether (m, σ, b) appears in the Σ-List. If not, $\mathcal{B}$ adds this signature σ in the $\overline{\Sigma}$-List and outputs `Invalid`. Otherwise $\mathcal{B}$ computes $R = P' - mS$ and $Q = P_b + ra_bP'$ where $\sigma = (S, r)$ and simulates the proof of equality $\log_S(R) = \log_{P'}(Q)$ as follows : $\mathcal{A}$ sends a commitment C to $\mathcal{B}$, who picks $\gamma \in [\![1, q-1]\!]$ at random and computes $R_1 = \gamma P'$ and $R_2 = \gamma Q$ and sends (R_1, R_2) to $\mathcal{A}$ who decommits (u, v) and $\mathcal{B}$ verifies that $C = uP' + vR$. If it is the case, $\mathcal{B}$ rewinds $\mathcal{A}$, and resets the simulation with the same random tape. He replays the same simulation up to the moment where $\mathcal{A}$ sends C. $\mathcal{B}$, now that he knows u, v, and y, picks $r \in [\![1, q-1]\!]$ at random and computes $R_1 = C + rP_\mathcal{A}$ (where $P_\mathcal{A} = yP'$) and $R_2 = (u + yr)Q + vS$, and sends it to $\mathcal{A}$, which accepts the proof.

When the adversary queries the denying oracle $\Upsilon_{D,b}$ with $b \in \{0,1\}$ on a putative signature $\sigma = (S, r)$ on m, $\mathcal{B}$ verifies that σ does not appear in the Σ-List (if it does it outputs `Valid`) and updates the $\overline{\Sigma}$-List with σ. Then $\mathcal{B}$ computes $R = P' - mS$ and $Q = P_b + ra_bP'$ and simulates the denying protocol as follows : when $\mathcal{A}$ sends (C, C') to him, $\mathcal{B}$ picks randomly $\tilde{r} \in [\![0, \lambda]\!]$ and sends to $\mathcal{A}$ the point $C'' = \tilde{r}P'$. $\mathcal{A}$ sends him back u, and $\mathcal{B}$ looks for $s' \in [\![0, \lambda]\!]$ such that $C = uP' + s'R$. If he does not find such an s', he aborts. Otherwise $\mathcal{B}$ computes $r = (\tilde{r} - s')y^{-1}$ so that $C'' = s'P' + rP_\mathcal{A}$ (where $P_\mathcal{A} = yP'$). Then $\mathcal{B}$ sends (s', r).

This simulation is perfect, therefore we have $\Pr[\mathsf{Guess}_4] = \Pr[\mathsf{Guess}_3]$.

Game_5 Now $\mathcal{B}$ simulates the challenge signature. The euclidean division of $f(y)$ by $(y+h)$ gives $\dfrac{f(y)}{y+h} = \dfrac{\gamma}{x+h} + \displaystyle\sum_{i=0}^{q_\Sigma - 2} \gamma_i y^i$. $\mathcal{B}$ picks at random $b^\star \in \{0,1\}$ and the challenge signature is $(S^\star, r^\star)$ where $S^\star = \dfrac{\gamma}{\alpha^{b^\star}} Q + \displaystyle\sum_{i=0}^{q_\Sigma - 2} \gamma_i \alpha^{b^\star(i-1)} A_i$ and $r^\star = (h\alpha^{b^\star} - m^\star)a_{b^\star}^{-1}$ which is likely to be zero with probability at most 2^{-k}. If this happens $\mathcal{B}$ aborts the simulation, otherwise he feds $\mathcal{A}$ with $(S^\star, r^\star)$. This game perfectly simulates the signing oracle unless it aborts.

This completes the description of $\mathcal{B}$. If $Q = Q_{\mathsf{real}} = (x+h)^{-1}P$, this game perfectly simulates the challenge generation if the event Bad does not occur and $\mathcal{B}$ does not abort (which happens with probability at most 2^{-k}). Therefore $|\Pr[\mathsf{Guess}_5 | Q = Q_{\mathsf{real}}] - \Pr[\mathsf{Guess}_4]| \leq 2^{-k} + \Pr[\mathsf{Bad}]$.

If $Q = Q_{\mathsf{random}}$ is a random element from $\mathbb{G}$, the adversary gains no information on b, in an information theoretic sense, therefore:
$\Pr[\mathsf{Guess}_5 | Q = Q_{\mathsf{random}}] \leq 1/2 + 2^{-k} + 2^{-k}$.

The last term accounts for the probability that $Q_{\mathsf{random}} = Q_{\mathsf{real}}$. By definition, the advantage in the Game_5 simulation in solving the $(q_\Sigma + 1)$-DSDH problem is: $\mathbf{Adv}^{(\mathsf{q}_\Sigma+1)-\mathsf{dsdh}}_{\mathsf{Gen},\mathsf{Game}_5}(k) = |\Pr[\mathsf{Guess}_5 | Q = Q_{\mathsf{real}}] - \Pr[\mathsf{Guess}_5 | Q = Q_{\mathsf{random}}]|$.
A simple computation gives the claimed bounds for ε' and τ'. □

Proposition 2. *Let Gen be a POGG and let US be the associated undeniable signature scheme. For any Ano-CMA adversary $\mathcal{A}$ against US, with security parameter k, which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{ano-cma}}_{US,\mathcal{A}}(k)$, running time τ, making q_Σ queries to the signing oracle, q_C to the confirming oracle, q_D to the denying oracle and registers up to $q_\mathcal{R}$ keys, there exists an EF-CMA-adversary $\mathcal{C}$ with success $\varepsilon'' = \mathbf{Succ}^{\mathsf{ef-cma}}_{US,\mathcal{A}}(k)$, running time τ'', making q_Σ queries to the signing oracle, q_C to the confirming oracle, q_D to the denying oracle and registers up to $q_\mathcal{R}$ keys such that $\varepsilon'' \geq \Pr[Bad_\mathcal{A}]$ and $\tau'' \leq \tau + O(1)$.*

Proof. Let k be a security parameter, $(q, \mathbb{G})$ be a couple generated by Gen. We consider random public key pk and we construct a simulation which produces an existential forgery associated to pk.

Game_0 Exactly the same game as in the previous proof. By definition, we still have $|2\Pr[\mathsf{Guess}_0] - 1| = \mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{US},\mathcal{A}}(k)$.

Game_1 In this game, the algorithm $\mathcal{C}$ simulates $\mathcal{A}$'s access to the oracles this way. It forwards $pk_0 = pk$ to $\mathcal{A}$ with a new public key pk_1 randomly reducible to pk (as in $\mathsf{Game}_?$? of the previous proof). $\mathcal{C}$ simulates $\mathcal{A}$'s signing and confirming/denying protocols by using its own signing and confirming/denying oracles for each of $\mathcal{A}$'s query. During the simulation, $\mathcal{C}$ stored in a $\overline{\Sigma}$-List any pair message/signature accepted by the confirming oracle, not obtained from his signing oracle.

At the end of its Find stage, $\mathcal{A}$ produces a message $m^\star$ and sends it to its challenger. $\mathcal{C}$ simulates this challenger by picking at random a bit $b^\star$ and produces either a real signature of $m^\star$ thanks to its signing oracle. Let $\sigma^\star$ be this signature. $\mathcal{C}$ sends $\sigma^\star$ to $\mathcal{A}$, who begins its Guess stage. The simulation of all oracles is the same as in the Find stage. Finally $\mathcal{A}$ produces a bit $b^\star$.

This game is clearly identical to the previous one. Hence, we obtain $\Pr[\mathsf{Guess}_1] = \Pr[\mathsf{Guess}_0]$.

Finally, $\mathcal{A}$'s output bit is discarded by $\mathcal{C}$, which outputs an element of the $\overline{\Sigma}$-List if it is not empty, and a random element of $\{0,1\}^* \times \mathbb{G} \times [\![1, q-1]\!]$ otherwise. Its running time is the same as $\mathcal{A}$'s, and its success it at least $\Pr[\mathsf{Bad}]$. □

Theorem 1 (Anonymity of US). *Let Gen be a POGG and let US be the associated undeniable signature scheme. For any Ano-CMA adversary $\mathcal{A}$ against US, with security parameter k, which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{ano-cma}}_{US,\mathcal{A}}(k)$, running time τ, making q_Σ queries to the signing oracle, q_C to the confirming oracle, q_D to the denying oracle and registers up to $q_\mathcal{R}$ keys, there exist*

- *an adversary $\mathcal{B}$ against $(q_\Sigma+1)$-DSDH of advantage $\varepsilon' = \mathbf{Adv}^{(\mathsf{q}_\Sigma+1)-\mathsf{dsdh}}_{Gen,\mathcal{B}}(k)$ and running time τ';*
- *an EF-CMA-adversary $\mathcal{C}$ with success $\varepsilon'' = \mathbf{Succ}^{\mathsf{ef-cma}}_{US,\mathcal{A}}(k)$, running time τ'', making q_Σ queries to the signing oracle, q_C to the confirming oracle, q_D to the denying oracle and registers up to $q_\mathcal{R}$ keys*

such that $\varepsilon' + \varepsilon'' \geq \varepsilon/2 - (q_\Sigma + 2)2^{-k}$, $\tau' \leq q_C\tau + q_\Sigma(q_\Sigma T_\mathbb{G} + O(1))$ and $\tau'' \leq \tau + O(1)$ where $T_\mathbb{G}$ denotes the time complexity to perform a scalar multiplication in $\mathbb{G}$.

Proof. The result is an obvious consequence of the two previous propositions. □

Unforgeability.

Theorem 2 (Unforgeability of US). *Let Gen be a POGG and let US be the associated undeniable signature scheme. Let $\mathcal{A}$ be an EF-CMA-adversary against US with success $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{US,\mathcal{A}}(k)$ within time τ making q_Σ queries to the signing oracle, q_C to the confirming oracles and q_D to the denying oracle.*

1. *There exists an adversary $\mathcal{B}$ against $(q_\Sigma + 1)$-GSDH of advantage $\varepsilon' = \mathbf{Adv}^{(\mathsf{q}_\Sigma+1)-\mathsf{gsdh}}_{Gen,\mathcal{B}}(k)$ with running time τ' such that:*
   $\varepsilon' \leq \varepsilon/2 - q_\Sigma 2^{-k}$ and $\tau' \leq \tau + (q_C + q_D)T_{\mathcal{DSDH}} + O(q_\Sigma)$
   where $T_{\mathcal{DSDH}}$ denotes the time complexity of the oracle $\mathcal{DSDH}$.
2. *There exists an adversary $\mathcal{C}$ against $(q_\Sigma + 1)$-CSDH of advantage $\varepsilon'' = \mathbf{Adv}^{(\mathsf{q}_\Sigma+1)-\mathsf{csdh}}_{Gen,\mathcal{C}}(k)$ with running time τ'' such that:*
   $\varepsilon'' \leq \varepsilon(2(q_C + q_D + 1))^{-1} - q_\Sigma 2^{-k}$ and $\tau'' \leq \tau + O(q_\Sigma)$.

Proof.(Sketch) We consider an EF-CMA-adversary $\mathcal{A}$ with success $\mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{US},\mathcal{A}}(k)$ within time τ. The key generation algorithm is run to produces a pair of keys (pk, sk). The adversary $\mathcal{A}$ is fed with pk, and, querying the signing oracle Σ, and the confirming and denying oracles Υ_C and Υ_D, outputs a couple message/signature $(m^\star, \sigma^\star)$, where $\sigma^\star$ was not obtained from the signing oracle. We denote by q_Σ the number of queries to the signing oracle and q_C to the confirming oracle and q_D to the denying oracle.

As in Boneh and Boyen proof of security, we will construct an algorithm $\mathcal{B}$ (*resp.* $\mathcal{B}$) which is likely to break the random instance of $(q_\Sigma + 1)$-GSDH (*resp.* the $(q_\Sigma + 1)$-CSDH): $(P, xP, x^2P, \dots, x^{q_\Sigma+1}P)$.

We distinguish two type of forgers. The simulation of any interaction with the adversary is indistinguishable from the real attack. The only difference comes from the possibility offered to the adversary to query a confirming/denying oracle on any couple message/signature of his choice.

1. Thanks to the decisional oracle associated to $(q_\Sigma + 1)$-SDH and the points $P, xP, x^2P, \dots, x^{q_\Sigma+1}P$, $\mathcal{B}$ can construct a static oracle $\mathcal{O}_x$, which, given Q and R as inputs, answers whether $R = xQ$. Therefore, $\mathcal{B}$ an perfectly simulate an appropriate proof as in the proof of the theorem 1. The rest of the simulation follows *mutatis mutandis* the one of Boneh and Boyen [1] from which we obtain the claimed bound on ε' and τ' once taken into account the computational cost of the simulation of the confirming and denying oracles.
2. We can suppose without loss of generality that the potential forgery output by $\mathcal{A}$ is queried to the confirming oracle at the end of $\mathcal{C}$'s execution. We say that a couple message/signature is *special* if it is a valid message/signature pair queried by $\mathcal{A}$ to the confirming oracle or the denying oracle such that the signature has not been obtained from the signing oracle (in particular $(m^\star, \sigma^\star)$ is special if $\mathcal{A}$ succeeds). $\mathcal{C}$ picks at random an index

$\ell \in [\![1, q_C + q_D + 1]\!]$ as its guess of first query of a special message/signature couple. In $\mathcal{A}$'s execution, we denote by s the actual index of this first query (and $s = \infty$ if $\mathcal{A}$ does not make such a request). For the i-th query with $i < \ell$, $\mathcal{C}$ chooses to confirm the signature if it has been made by the signing oracle and to deny it otherwise. This simulation is done as in the previous proof. If the ℓ-th query (m_ℓ, σ_ℓ), has been obtained from the signing oracle, then $\mathcal{C}$ aborts. Otherwise following *mutatis mutandis* Boneh and Boyen's simulation $\mathcal{C}$ tries to solve the $(q_\Sigma + 1) - CSDH$ problem using the value σ_ℓ. $\mathcal{C}$ does not abort with probability $1/(q_C + q_D + 1)$ and we get the bounds on ε'' and τ'' once taken into account the computational cost of this simulation. □

Corollary 1 (Security of US). *Let Gen be a POGG and US be the associated undeniable signature scheme. Under the DSDH assumption in Gen, US is EF-CMA and Ano-CMA secure against polynomial-time adversaries.*

4 Conclusion

We designed the first efficient construction for undeniable signatures. It is a variant of Boneh-Boyen's signature scheme in a situation where the DDH problem is supposed to be difficult. The unforgeability and the anonymity are related to variants of the strong Diffie-Hellman assumption. The new scheme offers the advantage of issuing short signatures. Moreover, the computational costs for the signer in the signature generation, the confirmation/denial protocols and the receipt generation algorithms are the lowest of all known schemes.

Zhang and Chen [24] proposed very recently a new digital signature scheme in a bilinear setting whose resistance to forgery is reduced, in the standard security model, to a new algorithmic problem called the *k-square roots problem*. This protocol is very close to Boneh and Boyen's scheme, the underlying non-linear operation in $[\![1, q-1]\!]$ being the square root extraction, instead of the inversion. The computational costs of generation and verification and the size of these signatures are identical to those of Boneh-Boyen's signatures. By embedding this scheme in a classical cryptographic setting we can construct, with the same technique, a new efficient undeniable signature scheme which can be proved unforgeable in the standard security model.

References

1. D. Boneh, X. Boyen: Short Signatures Without Random Oracles. Proc. of Eurocrypt'04, Springer LNCS Vol. 3027, 56–73 (2004)
2. D. Boneh, B. Lynn, H. Shacham: Short Signatures from the Weil Pairing. J. Cryptology 17 (4), 297–319 (2004)
3. J. Boyar, D. Chaum, I. B. Damgård, T.P. Pedersen: Convertible Undeniable Signatures. Proc. of Crypto'90, Springer Vol. LNCS 537, 189–205 (1991)
4. G. Brassard, D. Chaum, C. Crpeau: Minimum Disclosure Proofs of Knowledge. J. Comput. Syst. Sci., 37 (2) 156–189 (1988)

5. D. Chaum: Zero-Knowledge undeniable signatures. Proc. of Eurocrypt'90, Springer LNCS Vol. 473, 458–464 (1991)
6. D. Chaum, H. van Antwerpen: Undeniable Signatures. Proc. of Crypto'89, Springer LNCS Vol. 435, 212–216 (1990)
7. S. Even, O. Goldreich, S. Micali: On-Line/Off-Line Digital Signatures. J. Cryptology, 9 (1), 35–67 (1996)
8. S. Galbraith, W. Mao: Invisibility and anonymity of undeniable and confirmer signatures. Proc. of CT-RSA 2003, Springer LNCS Vol. 2612 80–97 (2003)
9. S. Galbraith, W. Mao, K.G. Paterson: RSA-based undeniable signatures for general moduli. Proc. of CT-RSA 2002, Springer LNCS Vol. 2271, 200–217 (2002)
10. R. Gennaro, T. Rabin, H. Krawczyk: RSA-based undeniable signatures. J. Cryptology 13 (4), 397–416 (2000)
11. S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. SIAM J. Computing, 17 (2), 281–308 (1988)
12. S. Goldwasser, E. Waisbard: Transformation of Digital Signature Schemes into Designated Confirmer Signature Schemes. Proc. of TCC'04, Springer LNCS Vol. 2951, 77–100 (2004)
13. M. Jakobsson, K. Sako, R. Impagliazzo: Designated Verifier Proofs and their Applications. Proc.of Eurocrypt'96, Springer LNCS Vol. 1070, 142–154 (1996)
14. K. Kurosawa, S.-H. Heng: 3-Move Undeniable Signature Scheme. Proc of Eurocrypt'05, Springer LNCS Vol. 3494, 181–197 (2005)
15. F. Laguillaumie, D. Vergnaud: Time-Selective Convertible Undeniable Signatures. Proc. of CT-RSA'05, Springer LNCS Vol. 3376, 154-171 (2005)
16. B. Libert, J.-J. Quisquater: Identity Based Undeniable Signatures. Proc. of CT-RSA 2004, Springer LNCS Vol. 2964, 112–125 (2004)
17. M. Michels, H. Petersen, P. Horster: Breaking and repairing a convertible undeniable signature scheme. Proc. of ACM Conference on Computer and Communications Security 1996, 148–152, ACM Press (1996)
18. J. Monnerat, S. Vaudenay: Generic Homomorphic Undeniable Signatures. Proc. of Asiacrypt'04, Springer LNCS Vol. 3329, 354–371 (2004)
19. J. Monnerat, S. Vaudenay: Undeniable Signatures Based on Characters: How to Sign with One Bit. Proc. of PKC 2004, Springer LNCS Vol. 2947, 69–85 (2004)
20. T. Okamoto, D. Pointcheval: The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. Proc. of PKC 2001, Springer LNCS Vol. 1992, 104–118 (2001)
21. W. Ogata, K. Kurosawa, S.-H. Heng: The Security of the FDH Variant of Chaum's Undeniable Signature Scheme. Proc of PKC 2005, Springer LNCS Vol. 3386, 328–345 (2005)
22. C. P. Schnorr: Efficient Signature Generation by Smart Cards. J. Cryptology 4 (3), 161–174 (1991)
23. F. Zhang, R. Safavi-Naini, W. Susilo: An Efficient Signature Scheme from Bilinear Pairings and its Applications. Proc. of PKC 2004, Springer LNCS Vol. 2947, 277–290 (2004)
24. Z. Zhang, X. Chen: Yet Another Short Signatures Without Random Oracles from Bilinear Pairings. Cryptology ePrint Archive, Report 2005/203 (2005)