# Gradually Convertible Undeniable Signatures

## (Michels-Petersen-Horster Convertible Undeniable Signatures Revisited)

Laila El Aimani and Damien Vergnaud

b-it COSEC - Bonn/Aachen International Center for Information Technology
Computer Security Group, Dahlmannstr. 2, D-53113 Bonn, Germany
`{elaimani,vergnaud}@bit.uni-bonn.de`

**Abstract.** In 1990, Boyar, Chaum, Damgård and Pedersen introduced *convertible undeniable signatures* which limit the self-authenticating property of digital signatures but can be converted by the signer to ordinary signatures. Michels, Petersen and Horster presented, in 1996, an attack on the El Gamal-based seminal scheme of Boyar *et al.* and proposed a repaired version without formal security analysis. In this paper, we modify their protocol so that it becomes a generic one and it provides an advanced feature which permits the signer to universally convert *achronously* all signatures pertaining to a specific time period. We supply a formal security treatment of the modified scheme: we prove, in the generic group model, that the protocol is existentially unforgeable and anonymous under chosen message attacks, assuming new assumptions (though reasonable) on the underlying hash function.

## 1 Introduction.

In 1996, Michels, Petersen and Horster [16] proposed an efficient convertible undeniable signature protocol whose security relies on the difficulty of the discrete logarithm problem in the multiplicative group of a finite field. This scheme has received little attention from the cryptographic community whereas we are convinced that it deserves better than oblivion. This article focuses on the security treatment and on the proposal of an additional functionality for Michels-Petersen-Horster convertible undeniable signatures. Our analysis points out new security properties for the underlying hash functions which may be of independent interest.

**Related work.** A property of conventional digital signature schemes is that once a signature is released, everybody can check its validity. However there are numerous situations where this *self-authenticating* property is not desirable. In 1989 Chaum and van Antwerpen [8] introduced the concept of *undeniable signatures* whose purpose is to perform public key digital signatures which cannot be verified without interacting with the signer. In addition to the confidentiality and privacy concerns in themselves, this primitive finds applications in such different fields as electronic payment systems, certificate management or cyberdemocracy.

In 1991, the concept has been refined by giving the possibility to transform an undeniable signature into an ordinary digital signature. These *convertible undeniable signatures*, proposed in [3] by Boyar, Chaum, Damgård and Pedersen, provide individual and universal conversions of the signatures. Unfortunately, this El Gamal-like scheme has been broken in 1996 by Michels, Petersen, and Horster [16] who proposed a repaired version with heuristic security.

The universal conversion of all convertible undeniable signature protocols proposed before 2005, consists in revealing a part of the signer's secret key. This conversion makes all signatures, *past as well as future*, be universally verifiable. This property may be undesirable in some context since the corresponding keys cannot be used to generate undeniable signatures any more. To overcome this problem, Laguillaumie and the second author introduced and formalized, in 2005 [14], the *time-selective convertible undeniable signatures* which supports signers in gradually converting the undeniable signatures in a controlled fashion. They proposed a scheme which permits the signer to universally convert *chronologically* signatures pertaining only to a specific time period: given a time-selective convertible undeniable signature $\sigma$ for a time period $t$, it is computationally infeasible to determine which signing secret key was used to generate $\sigma$; but with the knowledge of a matching universal receipt for some time period $p' \geq p$, it is easy to determine whether $\sigma$ is a valid time-selective convertible undeniable signature or not. A tantalizing challenge is to generalize the concept of time-selective convertible signature to *event-selective* convertible signature where a signature becomes universally verifiable if a specific event happens that makes the signer publish the corresponding receipt information. This primitive will enable the signer to gradually convert signatures *achronously* (*i.e.* with time periods made completely independent of each other). Up to now, no concrete realization of this concept has been proposed in the literature.

**Our contributions.** In this paper, we revisit the Michels-Petersen-Horster convertible undeniable signature scheme. First of all, we modify it such that it becomes a generic algorithm. This point of view allows to look at cryptographic constructions in an abstract way and "move" them to other groups without the original restriction of subgroup of the multiplicative group of a finite field. In addition, we suggest a slight modification of this scheme which gives the first realization of *achronous* gradually convertible undeniable signatures.

The security of many cryptographic tools relies on assumptions about the hardness of certain algorithmic problems. Techniques from [20] suggest that it is highly improbable to reduce the security of the Michels-Petersen-Horster signatures to the discrete logarithm problem in the standard security model. Therefore, we investigate their security in the so-called *generic group model*, following previous work from [5, 23] where the security of a generic version of the protocol DSA was analyzed. However, it is worth noting that the real, non-generic security of the scheme may be completely different in different groups [9].

This security analysis points out new sufficient security properties for the underlying hash functions. These new notions of *random affine preimage resistance* and *random linear collision resistance* are satisfied by generic hash functions (*i.e.* in the random oracle model [1]). The former property is necessary for our scheme to be secure, while the latter is necessary for instance for the RSA-FDH signature scheme [2].

**Notations.** The set of $n$-bit strings is denoted by $\{0,1\}^n$ and the set of all finite binary strings (or messages) is denoted by $\{0,1\}^*$. Concatenation of two strings $x$ and $y$ is denoted by $x\|y$. Let $\mathcal{A}$ be a probabilistic Turing machine running in polynomial time (a PPTM, for short), and let $x$ be an input for $\mathcal{A}$. The probability space that assigns to a string $\sigma$ the probability that $\mathcal{A}$, on input $x$, outputs $\sigma$ is denoted by $\mathcal{A}(x)$. The support of $\mathcal{A}(x)$ is denoted by $\mathcal{A}[x]$. Given a probability space $S$, a PPTM that samples a random element according to $S$ is denoted by $x \overset{R}{\longleftarrow} S$. For a finite set $X$, $x \overset{R}{\longleftarrow} X$ denotes a PPTM that samples a random element uniformly at random from $X$. A *two-party protocol* is a pair of interactive PPTMs $(\mathsf{Prove}, \mathsf{Verify})$.

## 2 Gradually convertible undeniable signatures

In this section, we formalize the concept of gradually convertible undeniable signatures.

### 2.1 Definition

As in ordinary digital signatures, undeniable signature schemes establish two complimentary algorithms: one for signing ($\mathsf{Sign}$) and the other for checking the signature at some later time ($\mathsf{Cont}$), but this algorithm is not publicly available since it requires the knowledge of the signer's secret key to be executed. Besides, the signer can prove his authorship of an undeniable signature by running a confirmation protocol ($\mathsf{Conf}$) with a verifier and a falsely implicated signer may deny his involvement by running a denial protocol ($\mathsf{Deny}$) with a verifier.

*Designated verifier proofs* were introduced by Jakobsson, Sako and Impagliazzo in 1996 [12] and have been widely used as non-transferable confirmation and denial protocols for undeniable signature schemes. In [13], Kudla and Paterson present a security model for these signatures where the confirmation and denial protocols are actually implemented with such proofs. They proposed non-interactive designated verifier proofs suited to combination with Chaum-van Antwerpen original undeniable signature scheme resulting in a secure[1] and efficient undeniable signature scheme. Unfortunately, we cannot use these non-interactive non-transferable proofs, to obtain the security results without the random oracle model. Indeed, as far as we know, all the non-interactive proofs are either highly inefficient or obtained by applying the Fiat-Shamir heuristic to interactive designated verifier proofs. Therefore, in this paper, we will use interactive version of the designated verifier proofs described in [13].

[1] in the random oracle model, assuming the intractability of the decisional Diffie-Hellman problem in the underlying group [19, 18, 10].

In addition, the signer has at its disposal one algorithm ($\mathsf{Convert}$) which permits to

- convert a given undeniable signature into a regular, universally verifiable signature. This operation does not affect other undeniable signatures.
- publish a universal trapdoor relative to a specific time period $p$ by the means of which all undeniable signatures for the time period $p$ become universally verifiable. The trapdoor has no impact whatsoever on undeniable signatures pertaining to a time period[2] $p' \neq p$.

The verification of the converted signatures is performed thanks to the algorithm $\mathsf{Vf}$.

**Definition 1 (Gradually Convertible Undeniable Signature).** *Let $\pi \in \mathbb{N}$. A* gradually convertible undeniable signature scheme *with $\pi$ time periods $\Sigma$ is a 9-tuple $\Sigma = (\mathsf{Setup}, \mathsf{SKg}, \mathsf{VKg}, \mathsf{Sign}, \mathsf{Cont}, \mathsf{Conf}, \mathsf{Deny}, \mathsf{Conv}, \mathsf{Vf})$ such that:*

- *$\Sigma.\mathsf{Setup}$, the* common parameter generation algorithm, *is a PPTM which takes an integer $k$ as input. The output are the* public parameters *$\mathcal{P}$. $k$ is called the* security parameter.
- *$\Sigma.\mathsf{SKeyGen}$, the* signer key generation algorithm, *is a PPTM which takes the public parameters as input. The output is a pair $(\mathbf{sk_s}, \mathbf{pk_s})$ where $\mathbf{sk_s}$ is called a* signing secret key *and $\mathbf{pk_s}$ a* signing public key.
- *$\Sigma.\mathsf{VKeyGen}$, the* verifier key generation algorithm, *is a PPTM which takes the public parameters as input. The output is a pair $(\mathbf{sk_v}, \mathbf{pk_v})$ where $\mathbf{sk_v}$ is called a* verifying secret key *and $\mathbf{pk_v}$ a* verifying public key.
- *$\Sigma.\mathsf{Sign}$, the* signing algorithm, *is a PPTM which takes the public parameters, a message, an integer in $[\![1, \pi]\!]$ and a signing secret key as inputs and outputs a bit string.*
- *$\Sigma.\mathsf{Cont}$, the* controlling algorithm, *is a PPTM which takes the public parameters, a message $m$, a bit string $\sigma$, an integer $p \in [\![1, \pi]\!]$ and a signing key pair $(\mathbf{sk_s}, \mathbf{pk_s})$ as inputs and outputs a bit. If the bit output is 1 then the bit string $\sigma$ is said to be a* signature *on $m$ for $\mathbf{pk_s}$ for the time period $p$.*
- *$\Sigma.\{\mathsf{Conf}.\mathsf{Deny}\}$, the* confirming/denying protocols *(respectively), are two-party protocols $(\mathsf{Prove}, \mathsf{Verify})$ such that:*
  - *$\mathsf{Prove}$ and $\mathsf{Verify}$ take as input a message $m$, an integer $p \in [\![1, \pi]\!]$, a bit-string $\sigma$, a signing public key $\mathbf{pk_s}$ and a verifying public key $\mathbf{pk_v}$ and the public parameters;*
  - *$\mathsf{Prove}$ takes as input $\mathbf{sk_s}$ the signing secret key corresponding to $\mathbf{pk_s}$;*
  - *$\mathsf{Verify}$ takes as input $\mathbf{sk_v}$ the verifying secret key corresponding to $\mathbf{pk_v}$;*

  *$\mathsf{Conf}.\mathsf{Verify}$ (*resp. *$\mathsf{Deny}.\mathsf{Verify}$ ) outputs an element in $\{\perp, 1\}$ (*resp. *$\{\perp, 0\}$).*
- *$\Sigma.\mathsf{Conv}$, the* conversion algorithm, *is a PPTM which takes as input the public parameters, an integer $p \in [\![1, \pi]\!]$, a signing key pair $(\mathbf{sk_s}, \mathbf{pk_s})$ and a bit string $\Upsilon$.*

[2] In time-selective convertible undeniable signatures from [14], these universal receipts makes it possible to universally verify all signature for any time period $p' \leq p$

- *If $\Upsilon$ consists of a message $m$ and a signature $\sigma$ on $m$ for $\mathbf{pk_s}$ for the time period $p$, then it outputs a bit string $\tilde{\sigma}$;*
- *if $\Upsilon$ is the empty string $\varepsilon$, then it outputs a bit string $\mathcal{I}_p$;*

– *$\Sigma.\mathsf{Vf}$, the* verifying algorithm for converted signature, *is a PPTM which takes as input the public parameters, a message $m$, and a bit string $\sigma$, an integer $p \in \llbracket 1, \pi \rrbracket$, a signing public key $\mathbf{pk_s}$ and a bit string $\Lambda$ and outputs a bit. If the bit output is $1$ then the bit string $\Lambda$ is said to be a* receipt *of the validity of $\sigma$.*

*where the protocols $\Sigma.\mathsf{Conf}$ and $\Sigma.\mathsf{Deny}$ are a designated verifier proof of membership system for the languages (respectively):*

$$\{(\mathcal{P}, m, \sigma, p, \mathbf{pk_s}) \in \Sigma.\mathsf{Setup}[k] \times \{0,1\}^{*2} \times \llbracket 1, \pi \rrbracket \times \Sigma.\mathsf{SKg}[\mathcal{P}] | \Sigma.\mathsf{Vf}[\mathcal{P}, m, \sigma, p]\} = \{1\}$$
$$\{(\mathcal{P}, m, \sigma, p, \mathbf{pk_s}) \in \Sigma.\mathsf{Setup}[k] \times \{0,1\}^{*2} \times \llbracket 1, \pi \rrbracket \times \Sigma.\mathsf{SKg}[\mathcal{P}] | \Sigma.\mathsf{Vf}[\mathcal{P}, m, \sigma, p]\} = \{0\}$$

*and for all $k \in \mathbb{N}$, for all $\mathcal{P} \in \Sigma.\mathsf{Setup}[k]$, for all $\mathcal{S} = (\mathbf{pk_s}, \mathbf{sk_s}) \in \Sigma.\mathsf{SKg}[\mathcal{P}]$, for all $m \in \{0,1\}^*$ and for all $p \in \llbracket 1, \pi \rrbracket$, we have:*

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{sk_s}], \Sigma.\mathsf{Cont}[\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})] = \{1\}$$
$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{sk_s}], \forall \Lambda \in \Sigma.\mathsf{Conv}[\mathcal{P}, p, \mathcal{S}, (m, \sigma)], \Sigma.\mathsf{Vf}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\}$$
$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{pk_s}], \forall \Lambda \in \Sigma.\mathsf{Conv}[\mathcal{P}, p, \mathcal{S}, \varepsilon], \Sigma.\mathsf{Vf}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\}$$
$$\forall \sigma, \Lambda \in \{0,1\}^*, \Sigma.\mathsf{Vf}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\} \Rightarrow \Sigma.\mathsf{Cont}[\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})] = \{1\}.$$

*Remark 1.* The first two properties capture the validity and the non-transferable property of the protocols Conf and Deny (*i.e.* the use of designated verifier proofs insures that a verifier will gain no information in an execution of one of these protocols [13]). The latter properties are the properties of *correctness*:

– a well-formed signature is always accepted by the algorithm Cont;
– a receipt correctly constructed is always accepted by the algorithm Verify;
– and if there exists a bit-string $\Lambda$ which makes accepted a bit-string $\sigma$ by the algorithm Verify, then $\sigma$ is a valid signature.

### 2.2 Security model

**Registered public key model.** In public key cryptography, the notion of anonymity is to be handled with great attention. For instance, in order to ensure anonymity, it is important that users register their public key by a certifying authority. Hence, in our security analysis, it is assumed that the users' keys have been already registered to an authority. The registration procedure would always contain a proof of knowledge of the associated private key. To further simplify the security analysis, we will assume that this procedure will be the *direct registration of the keys*[3].

[3] It is often necessary to require the security of the schemes even if the adversary is the key registration center. In this case, one must replace the proof of knowledge associated to the key registration by a zero-knowledge one.

**Security against existential forgery under chosen message attack.** The standard notion of security for digital signatures was defined by Goldwasser, Micali and Rivest [11] as *existential forgery against adaptive chosen message attacks* (EF-CMA). In [14], the corresponding notion for time-selective convertible undeniable signatures is defined along the same lines. The natural definition of *resistance to forgery* for gradually convertible undeniable signatures that we propose is identical. In fact, we suppose that the adversary has access to the universal receipts $\Lambda_p$ for every time period $p \in [\![1, \pi]\!]$ and is allowed to query a converting oracle $\mathfrak{Cv}$, a confirming oracle $\mathfrak{C}$ amd a denying oracle $\mathfrak{D}$ on any couple message/signature of its choice. As usual, in the adversary answer, there is the natural restriction that the returned message/signature has not been obtained from the signing oracle.

**Definition 2 (Unforgeability - EF-CMA).** *Let $\pi$ be a positive integer, let $\Sigma = (\mathsf{Setup}, \mathsf{SKg}, \mathsf{VKg}, \mathsf{Sign}, \mathsf{Cont}, \mathsf{Conf}, \mathsf{Deny}, \mathsf{Conv}, \mathsf{Vf})$ be a gradually convertible undeniable signature scheme with $\pi$ time periods and let $\mathcal{A}$ be an PPTM. We consider the following random experiment, where k is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{ef-cma}}_{\Sigma,\mathcal{A}}(k)}$
$\mathcal{P} \overset{R}{\longleftarrow} \Sigma.\mathsf{Setup}(k)$,
$(\mathbf{pk_s}, \mathbf{sk_s}) \overset{R}{\longleftarrow} \Sigma.\mathsf{SKg}(\mathcal{P})$
for $j = 1$ to $\pi$ do $\Lambda_j \leftarrow \Sigma.\mathsf{Convert}(\mathcal{P}, \varepsilon, j, \mathbf{sk_s}, \varepsilon)$
$(m^\star, \sigma^\star, p^\star) \overset{R}{\longleftarrow} \mathcal{A}^{\mathfrak{S}, \mathfrak{Cv}, \mathfrak{C}, \mathfrak{D}}(\text{params}, pk, \{\Lambda_j\}_{j \in [\![1,\pi]\!]})$
    | $\mathfrak{S} : (m, p) \longrightarrow \Sigma.\mathsf{Sign}(\mathcal{P}, m, p, \mathbf{sk_s})$
    | $\mathfrak{Cv} : (m, p, \sigma) \longrightarrow \Sigma.\mathsf{Convert}(\mathcal{P}, p, (\mathbf{sk_s}, \mathbf{pk_s}), (m, \sigma))$
    | $\mathfrak{C} : (m, p, \sigma, \mathbf{pk_v}) \longrightarrow \Sigma.\mathsf{Conf}(m, p, \sigma, \mathbf{pk_v}, \mathbf{pk_s})$
    | $\mathfrak{D} : (m, p, \sigma, \mathbf{pk_v}) \longrightarrow \Sigma.\mathsf{Deny}(m, p, \sigma, \mathbf{pk_v}, \mathbf{pk_s})$
return 1 if and only if the following properties are satisfied:
    - $\Sigma.\mathsf{Verify}[\mathcal{P}, \mathbf{pk_s}, m^\star, \sigma^\star, \Lambda_{p^\star}] = \{1\}$
    - $m$ was not queried to $\mathfrak{S}$
```

*We define the* success *of $\mathcal{A}$,* via $\mathbf{Succ}^{\textit{ef-cma}}_{\Sigma,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\textit{ef-cma}}_{\Sigma,\mathcal{A}}(k) = 1\right]$.
*Given $(k, t) \in \mathbb{N}^2$ and $\varepsilon \in [0, 1]$, the scheme $\Sigma$ is said to be $(k, t, \varepsilon)$-EF-CMA secure, if no EF-CMA-adversary $\mathcal{A}$ running in time $t$ has $\mathbf{Succ}^{\textit{ef-cma}}_{\Sigma,\mathcal{A}}(k) \geq \varepsilon$. The scheme $\Sigma$ is said to be EF-CMA secure if, for any security parameter $k \in \mathbb{N}$, any polynomial function $t : \mathbb{N} \to \mathbb{N}$, and any negligible function $\varepsilon : \mathbb{N} \to [0, 1]$, it is $(k, t(k), \varepsilon(k))$-EF-CMA secure.*

**Anonymity.** We state the precise definition of *anonymity* under a chosen message attack (Ano-CMA) which captures the notion that an attacker cannot determine under which key a signature was performed [10]. We consider a Ano-CMA-adversary $\mathcal{A}$ that runs in two stages. In the find stage, it takes as input two signing public keys $\mathbf{pk_{s}}_0$ and $\mathbf{pk_{s}}_1$ and outputs a message $m^\star$, a time period $p^\star$ together with some state information $\mathcal{I}$. In the guess stage, $\mathcal{A}$ gets a challenge gradually convertible undeniable signature $\sigma^\star$ formed by signing at random the message $m^\star$ under one of the two keys for the time period $p^\star$ and it must say which key was chosen. In both stages, the adversary has access to a signing oracle $\mathfrak{S}$ for both signing key pairs, to a converting oracle $\mathfrak{Cv}$, to a confirming oracle $\mathfrak{C}$ and to a denying oracle $\mathfrak{D}$. The attacker is also given the universal receipts of both potential signers for all[4] time period $p \in [\![1,\pi]\!] \setminus \{p^\star\}$. The only restriction on $\mathcal{A}$ is that it cannot query the triple $(m^\star, \sigma^\star, p^\star)$ on the converting and confirming/denying oracles.

**Definition 3 (Anonymity - Ano-CMA).** *Let $\pi$ be a positive integer, let $\Sigma = (\mathsf{Setup}, \mathsf{SKg}, \mathsf{VKg}, \mathsf{Sign}, \mathsf{Cont}, \mathsf{Conf}, \mathsf{Deny}, \mathsf{Conv}, \mathsf{Vf})$ be a gradually convertible undeniable signature scheme with $\pi$ time periods and let $\mathcal{A}$ be an PPTM. We consider the following random experiment, for $r \in \{0,1\}$, where $k$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\textsf{ano-cma}-r}_{\Sigma,\mathcal{A}}(k)}$
$\mathcal{P} \xleftarrow{R} \Sigma.\mathsf{Setup}(k)$
$(\mathbf{pk}_{\mathbf{s}0}, \mathbf{sk}_{\mathbf{s}0}) \xleftarrow{R} \Sigma.\mathsf{SKeyGen}(\mathcal{P})$,
$(\mathbf{pk}_{\mathbf{s}1}, \mathbf{sk}_{\mathbf{s}1}) \xleftarrow{R} \Sigma.\mathsf{SKeyGen}(\mathcal{P})$
$(m^\star, p^\star, \mathcal{I}) \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{Cv},\mathfrak{C},\mathfrak{D}}(\mathsf{find}, \mathcal{P}, \mathbf{pk}_{\mathbf{s}0}, \mathbf{pk}_{\mathbf{s}1})$
    | $\mathfrak{S} : (m, p, i) \longrightarrow \Sigma.\mathsf{Sign}(\mathcal{P}, m, p, \mathbf{sk}_{\mathbf{s}i})$
    | $\mathfrak{Cv} : (m, p, \sigma, i) \longrightarrow \Sigma.\mathsf{Convert}(\mathcal{P}, p, (\mathbf{sk}_{\mathbf{s}i}, \mathbf{pk}_{\mathbf{s}i}), (m, \sigma))$
    | $\mathfrak{C} : (m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, i) \longrightarrow \Sigma.\mathsf{Conf}(m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, \mathbf{pk}_{\mathbf{s}i})$
    | $\mathfrak{D} : (m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, i) \longrightarrow \Sigma.\mathsf{Deny}(m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, \mathbf{pk}_{\mathbf{s}i})$
$\sigma^\star \xleftarrow{R} \Sigma.\mathsf{Sign}(\mathcal{P}, m, \mathbf{sk}_{\mathbf{s}r}, p^\star)$
$\mathsf{for}\ j\ \mathsf{from}\ 1\ \mathsf{to}\ \pi\ \mathsf{do}$
    $\Lambda_j^0 \leftarrow \Sigma.\mathsf{Convert}(\mathcal{P}, \varepsilon, \mathbf{pk}_{\mathbf{s}0}, \mathbf{sk}_{\mathbf{s}0}, j)$ and $\Lambda_j^1 \xleftarrow{R} \Sigma.\mathsf{Convert}(\mathcal{P}, \varepsilon, \mathbf{pk}_{\mathbf{s}1}, \mathbf{sk}_{\mathbf{s}1}, j)$
$d \leftarrow \mathcal{A}^{\mathfrak{S},\mathfrak{Cv},\mathfrak{C},\mathfrak{D}}(\mathsf{guess}, \mathcal{I}, \{\Lambda_j^0, \Lambda_j^1\}_{j \in [\![1,\pi]\!] \setminus \{p^\star\}})$
Return $d$
```

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

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

*Given $(k,t) \in \mathbb{N}^2$ and $\varepsilon \in [0,1]$, the scheme $\Sigma$ is said to be $(k,t,\varepsilon)$-Ano-CMA secure, if no Ano-CMA-adversary $\mathcal{A}$ running in time $t$ has $\mathbf{Adv}^{\textsf{ano-cma}}_{\Sigma,\mathcal{A}}(k) \geq \varepsilon$. The scheme $\Sigma$ is said to be Ano-CMA secure if, for any security parameter $k \in \mathbb{N}$, any polynomial function $t : \mathbb{N} \rightarrow \mathbb{N}$, and any negligible function $\varepsilon : \mathbb{N} \rightarrow [0,1]$, it is $(k, t(k), \varepsilon(k))$-Ano-CMA secure.*

## 3 Hash functions and new security properties

Hash functions take messages of arbitrary length and outputs a fixed length string. In cryptographic uses of a hash function $\mathcal{H} : \{0,1\}^* \longrightarrow H$, these properties are considered prerequisites:

[4] This is the main difference with time-selective convertible undeniable signatures from [14] where this universal receipts was given only for $p \in [\![1, p^\star - 1]\!]$.

- *Preimage resistance*: given $h \in H$, it should be computationally intractable to find a message $m$ such that $\mathcal{H}(m) = h$.
- *Collision-resistant:* it should be computationnally intractable to find two different messages $m_1$ and $m_2$ such that $\mathcal{H}(m_1) = \mathcal{H}(m_2)$.

In this section, we formulate generalization of these security notions and study their properties.

**Definitions.** The proof of security of our variant of Michels-Petersen-Horster signatures makes use of new non-standard variations of the preimage resistance and the collision resistance assumptions for hash functions. These assumptions are of independent interest as they have interesting relations with the classical ones. We call them *random affine preimage resistance* and *random linear collision resistance.* Despite, being stronger than the standard assumptions, they are quite realistic.

According to [21], an hash function family is a family of functions $(\mathcal{H}_k : \mathcal{K}_k \times \{0,1\}^* \longrightarrow \{0,1\}^k)_{k\in\mathbb{N}}$, where $\mathcal{K}_k$ is a finite non-empty set. We will write the first argument of $\mathcal{H}_k$ as a subscript, so that $\mathcal{H}_{K,k}(m) = \mathcal{H}_k(K,m)$. In the following, we denote elements from $\{0,1\}^k$ as the corresponding $k$-bits integers in binary representation and we will denote for every integer $N \in \mathbb{Z}$, $\mathcal{H}^N_{K,k}$ the map defined by: $\mathcal{H}^N_{K,k} : \begin{cases} \{0,1\}^* \longrightarrow \mathbb{Z}_N \\ \quad m \longmapsto \mathcal{H}_{K,k}(m) \mod N. \end{cases}$

The new security definitions can be quantified as follows:

**Definition 4 (Random affine preimage resistance).** *Let $n$ be an integer, let $(\mathcal{H}_k : \mathcal{K}_k \times \{0,1\}^* \longrightarrow \{0,1\}^k)_{k\in\mathbb{N}}$ be an hash function family and let $\mathcal{A}$ be a PPTM. The success $\mathbf{Succ}^{raPre(n)}_{\mathcal{H},\mathcal{A}}(k)$ of $\mathcal{A}$ against the $n$-random affine preimage resistance of $\mathcal{H} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ is defined by:*

$$\max_{\substack{2^{k-1}\le N<2^k \\ \alpha_1,\dots,\alpha_n\in\mathbb{Z}^*_N \\ \beta_1,\dots,\beta_n\in\mathbb{Z}^*_N}} \left\{ \Pr\begin{bmatrix} K \xleftarrow{R} \mathcal{K}_k; (m,i,j) \xleftarrow{R} \mathcal{A}(K,\alpha_1,\dots,\alpha_n,\beta_1,\dots,\beta_n) \\ m \in \{0,1\}^*, (i,j) \in [\![1,n]\!]^2, i \neq j \\ \alpha_i + \beta_j \mathcal{H}^N_{K,k}(m) = 0 \mod N \end{bmatrix} \right\}.$$

An adversary $\mathcal{A}$ against the $n$-random affine preimage resistance of a hash function family $(\mathcal{H}_k)_{k\in\mathbb{N}}$ can be transformed easily into an adversary against the classical preimage resistance of $(\mathcal{H}_k)_{k\in\mathbb{N}}$ with success probability greater than $\mathbf{Succ}^{raPre(n)}_{\mathcal{H},\mathcal{A}}(k)/n^2$ and time-complexity of $\mathcal{A}$ increased by the time necessary to compute $n$ modular multiplications modulo $N$. In particular, the 1-random affine preimage resistance is equivalent to the classical preimage resistance.

**Definition 5 (Random linear collision resistance).** *Let $n$ be an integer, let $(\mathcal{H}_k : \mathcal{K}_k \times \{0,1\}^* \longrightarrow \{0,1\}^k)_{k\in\mathbb{N}}$ be an hash function family and let $\mathcal{A}$ be a PPTM. The success $\mathbf{Succ}^{rlColl(n)}_{\mathcal{H},\mathcal{A}}(k)$ of $\mathcal{A}$ against the $n$-random affine preimage* *resistance of* $\mathcal{H} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ *is defined by:*

$$\max_{\substack{2^{k-1}\leq N<2^k\\ \lambda_1,\ldots,\lambda_n\in\mathbb{Z}_N^*}} \left\{ \Pr \left[ \begin{array}{c} K \xleftarrow{R} \mathcal{K}_k; (m, m', i, j) \xleftarrow{R} \mathcal{A}(K, \lambda_1, \ldots, \lambda_n) \\ m, m' \in \{0,1\}^*, (i,j) \in \llbracket 1, n\rrbracket^2, m \neq m' \\ \lambda_i \cdot \mathcal{H}_{K,N}(m) = \lambda_j \cdot \mathcal{H}_{K,N}(m') \mod N \end{array} \right] \right\}.$$

As for random affine preimage resistance, the 1-random linear collision resistance is equivalent to the classical collision resistance. Unfortunately, it is impossible to prove that the $n$-random linear collision resistance can be reduced generically to the collision resistance for $n \geq 2$.

*Remark 2.* This security requirement is however reasonable since if the hash function family underlying the protocol RSA-FDH [2] does not satisfy it, then it is existential forgery against a *one* chosen-message attack: given an RSA public key $(N, e)$, the adversary can simply pick at random $r_1, \ldots, r_n \in \mathbb{Z}_N$, compute $\lambda_i = r_i^e \mod N$ for all $i \in \llbracket 1, n\rrbracket$, and try to find a random linear collision with parameters $N, \lambda_1, \ldots, \lambda_n$. If a collision $m, m' \in \{0,1\}^*, (i,j) \in \llbracket 1, n\rrbracket^2$, (such that $\lambda_i \cdot \mathcal{H}_{K,N}(m) = \lambda_j \cdot \mathcal{H}_{K,N}(m') \mod N$ is found), then the adversary queries the signature $\sigma$ on $m$ to the signing oracle and can compute the signature of $m'$ as $\sigma' = r_i \cdot \sigma \cdot r_j^{-1} \mod N$.

**Generic security** . The best known general collision-finding attack against a hash function family is the so-called birthday-attack. If we assume that the values of the hash-function family $(\mathcal{H}_k)_{k\in\mathbb{N}}$ are uniformly distributed over $\{0,1\}^k$ and that the generalisation of the birthday attack[5] against the random affine preimage resistance and the random linear collision resistance of $(\mathcal{H}_k)_{k\in\mathbb{N}}$ is the best possible attack (which is true in the random oracle model), then it is possible to give exponential lower bounds on the minimum of $n$ and of the number of hash functions evaluation required to have non-negligible probability of success. Indeed, for any integer $N \geq 2$, and for $(i,k) \in \mathbb{Z}_N$, it is straightforward [**?**] that

$$\#\{j \in \mathbb{Z}_N | i \cdot j \mod N \leq k\} = \gcd(i,N) \times \left( \left\lfloor \frac{k}{\gcd(i,N)} \right\rfloor + 1 \right).$$

Therefore if $D$ denotes the product of two independent random variables uniformly distributed over $\mathbb{Z}_N$, we have $\forall k \in \mathbb{Z}_N$

$$\Pr(D \leq k) = \frac{1}{N^2} \sum_{i=0}^{N-1} \gcd(i,N) \left( \left\lfloor \frac{k}{\gcd(i,N)} \right\rfloor + 1 \right),$$

and consequently, $D$ is close to the uniform distribution ove $\mathbb{Z}_N$. The results from [**?**] are sufficient to conclude; details will appear elsewhere.

[5] These attacks consist in picking messages $m_1, \ldots, m_r$, computing $h_i = \mathcal{H}_k(m_i) \mod N$ for $i \in \llbracket 1, r\rrbracket$ and $\gamma_{i,j} = -h_i\beta_j \mod N$ (*resp.* $\gamma_{i,j} = h_i\lambda_j \mod N$) for $j \in \llbracket 1, n\rrbracket$. They are successful if there is a triple $(i, j, \ell)$ $in\llbracket 1, r\rrbracket \times \llbracket 1, n\rrbracket^2$ (*resp.* a 4-tuple $(i, i', j, j') \in \llbracket 1, r\rrbracket^2 \times \llbracket 1, n\rrbracket^2$) s. t. $\gamma_{i,j} = \alpha_\ell$ (*resp.* $\gamma_{i,j} = \gamma_{i',j'}$ and $j \neq j'$).

# 4 Michels-Petersen-Horster convertible undeniable signatures revisited

## 4.1 Description of the scheme

Let $\pi$ be an integer. Following the notations in 2.1, the scheme can be described as follows:

- $\Sigma$.Setup: we consider a group $G$ of prime order $q$ generated by the element $P$, an encoding $\sigma : G \to S$, a reduction function[6] $F : S \to \mathbb{Z}_q$, a hash function $h : \{0,1\}^* \to \mathbb{Z}_q$ and two pseudo-random functions $H_1 : [\![1, \pi]\!] \times \mathbb{Z}_q \to \{0,1\}^k$ and $H_2 : \{0,1\}^* \times \{0,1\}^k \times S \to \mathbb{Z}_q$.
- $\Sigma$.SKg: $u, v \xleftarrow{R} [\![1, q-1]\!]$, compute $U \leftarrow uP$ and $V \leftarrow vP$.
- $\Sigma$.VKg: $w \xleftarrow{R} [\![1, q-1]\!]$, compute $W \leftarrow wP$.
- $\Sigma$.Sign: on message $m$ and period $p$, we do the following:
  - $r \xleftarrow{R} [\![1, q-1]\!]$, $R \leftarrow rP$. If $F(R) = 0$ we try with another value $r$.
  - $e_p \leftarrow H_1(p, v)$, $d \leftarrow H_2(m, e_p, R)$, $T \leftarrow dP$
  - $s \leftarrow (F(T) \cdot d \cdot h(m) \cdot v - u \cdot F(R) - 1)r^{-1} \bmod q$

  The signature is the tuple $(R, T, s)$.
- $\Sigma$.Cont: check that: $(v \cdot F(T) \cdot h(m)) \cdot T = F(R) \cdot U + s \cdot R + P$ using the private key $v$.
- $\Sigma$.{Conf/Deny}: the signer provides a designated verifier proof of knowledge of the equality/inequality of two discrete logarithms, namely, $F(R) \cdot U + s \cdot R + P$ to the base $(F(T).h(m))T$ and $V$ to the base $P$ (see appendix 4.2).
- $\Sigma$.Convert: we allow two types of conversions, namely
  - The gradual conversion for the signature corresponding to the time period $p$ could be done by releasing the value $e_p$.
  - The individual conversion can be achieved by releasing the value of $d$.
- $\Sigma$.Verify: The signature corresponding to the period $p$, once $e_p$ or $d$ is revealed, could be checked by any verifier using the equations: $(d \cdot F(T) \cdot h(m))V = F(R) \cdot U + s \cdot R + P$ and $T = dP$.

## 4.2 Proofs of equality/inequality of discrete logarithms

Let $\mathbb{G}$ be a group. To confirm or deny that a bit string is a signature in our undeniable signature scheme, it is necessary to prove that a given quadruple $(U_1, V_1, U_2, V_2) \in \mathbb{G}^4$ is a Diffie-Hellman quadruple (or not), *i.e.* belongs to the set

$$\mathsf{EDL}(\mathbb{G}) = \{(U_1, V_1, U_2, V_2) \in \mathbb{G}^4, \log_{U_1}(V_1) = \log U_2(V_2)\},$$

(or to the set $\mathsf{IDL}(\mathbb{G}) = \mathbb{G}^4 \setminus \mathsf{EDL}(\mathbb{G})$).

To face *blackmailing* or *mafia* attacks against our undeniable signatures, we use interactive designated verifier proofs, as introduced in [12] by Jakobsson, Sako, and Impagliazzo, in Chaum's proofs of equality (*cf.* Fig. 1) and inequality

[6] See 5.1

(*cf.* Fig. 2) of discrete logarithm of [6]. 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 protocols, involve a point $Y = yU_1$ where $y$ is the secret key of the verifier, and the prover must be convinced that $Y$ is well-formed (in the registered public key model, the registration protocol is used to force the users to know the secret-key corresponding to their public key).

```
Protocol EDL.Prove
Common input: $(U_1, U_2, V_1, V_2)$, $Y$
$\mathcal{P}$'s input: $x$
$\mathcal{V}$'s output: $b$
① $\mathcal{P} \xrightarrow{C_1, C_2, C_3} \mathcal{V}$
$(a, b, k) \overset{R}{\longleftarrow} [\![1, q-1]\!]^3$
$C_1 \leftarrow [k] \cdot U_1$ ; $C_2 \leftarrow [k] \cdot U_2$
$C_3 \leftarrow [a] \cdot U_1 + [b] \cdot Y$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \overset{R}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c} \mathcal{V}$
$c \leftarrow k - x(r + b) \mod q$

• $\mathcal{V}$'s execution ending
$\widetilde{C_1} \leftarrow [c] \cdot U_1 + [r + b] \cdot V_1$
$\widetilde{C_2} \leftarrow [c] \cdot U_2 + [r + b] \cdot V_2$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
if $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$
    then $b \leftarrow$ Accept else $b \leftarrow \perp$
```

```
Protocol EDL.Fake
Common input: $(U_1, U_2, V_1, V_2)$, $Y$
$\mathcal{P}$'s input: $y$
$\mathcal{V}$'s output: $b$
① $\mathcal{P} \xrightarrow{C_1, C_2, C_3} \mathcal{V}$
$(c, d, k) \overset{R}{\longleftarrow} [\![1, q-1]\!]^3$
$C_1 \leftarrow [c] \cdot U_1 + [d] \cdot V_1$ ; $C_2 \leftarrow [c] \cdot U_2 + [d] \cdot V_2$
$C_3 \leftarrow [k] \cdot U_1$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \overset{R}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c} \mathcal{V}$
$b \leftarrow d - r \mod q$ ; $a \leftarrow k - by \mod q$

• $\mathcal{V}$'s execution ending
$\widetilde{C_1} \leftarrow [c] \cdot U_1 + [r + b] \cdot V_1$
$\widetilde{C_2} \leftarrow [c] \cdot U_2 + [r + b] \cdot V_2$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
if $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$
    then $b \leftarrow$ Accept else $b \leftarrow \perp$
```

**Fig. 1.** Interactive designated verifier proof of membership of the language $\mathsf{EDL}(\mathbb{G})$

## 5 Security analysis

We note first that the property of non-transferability is fulfilled by our scheme as a direct consequence of the use of designated-verifier proofs in the confirm/deny protocols. Further, we state that our scheme resists existential forgeries and that signatures are anonymous. Both security reductions stand in the generic group model.

### 5.1 The generic group model

**The model.** A generic model of a group was first introduced by Nechaev [17]. Shoup [22] later improved these results and applied this model to cryptology. In

```
Protocol IDL.Prove
Common input: $(U_1, U_2, V_1, V_2)$, $Y$
$\mathcal{P}$'s input : $x$
$\mathcal{V}$'s output : $b$
① $\mathcal{P} \xrightarrow{C_0, C_1, C_2, C_3} \mathcal{V}$
$(a, b, k_0, k_1, k_2) \overset{R}{\longleftarrow} [\![1, q-1]\!]^5$
$C_0 \leftarrow [k_0] \cdot (V_2 - [x] \cdot U_2)$
$C_1 \leftarrow [k_1] \cdot U_1 - [k_2] \cdot V_1$
$C_2 \leftarrow [k_1] \cdot U_2 - [k_2] \cdot V_2$
$C_3 \leftarrow [a] \cdot U_1 + [b] \cdot Y$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \overset{R}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c, d} \mathcal{V}$
$c \leftarrow k_1 - xk_0(r + b) \mod q$
$d \leftarrow k_2 - k_0(r + b) \mod q$

• $\mathcal{V}$'s execution ending
$\widetilde{C_1} \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$\widetilde{C_2} \leftarrow C_0 + [c] \cdot U_2 - [r + b] \cdot V_2$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
if $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) \wedge C_0 \neq \mathbb{O}_{\mathbb{G}_2}$
    then $b \leftarrow$ Accept else $b \leftarrow \perp$
```

```
Protocole IDL.Fake
Common input: $(U_1, U_2, V_1, V_2)$, $Y$
$\mathcal{P}$'s input: $y$
$\mathcal{V}$'s output: $b$
① $\mathcal{P} \xrightarrow{C_0, C_1, C_2, C_3} \mathcal{V}$
$(c, d, k_1, k_2) \overset{R}{\longleftarrow} [\![1, q-1]\!]^4$
$C_0 \overset{R}{\longleftarrow} \mathbb{G} \setminus \{\mathbb{O}_{\mathbb{G}}\}$ ; $C_1 \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$C_2 \leftarrow C_0 + [c] \cdot U_2 - [k_1] \cdot V_2$
$C_3 \leftarrow [k_2] \cdot U_1$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \overset{R}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c, d} \mathcal{V}$
$b \leftarrow k_1 - r \mod q$ ; $a \leftarrow b - k_2 y \mod q$

• $\mathcal{V}$'s execution ending
$\widetilde{C_1} \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$\widetilde{C_2} \leftarrow C_0 + [c] \cdot U_2 - [r + b] \cdot V_2$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
if $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3}) \wedge C_0 \neq \mathbb{O}_{\mathbb{G}_2}$
    then $b \leftarrow$ Accept else $b \leftarrow \perp$
```

**Fig. 2.** Interactive designated verifier proof of membership to the language $\mathsf{IDL}(\mathbb{G})$

this model, called also the `black-box` model, one assumes that group operations in a group can be perfomed only by means of an oracle. More specifically, suppose that $G$ is an (additive) group of prime order $q$. Then $G$ is isomorphic to the additive group $\mathbb{Z}_q$ and for any non-identity element $P \in G$, one can construct an efficient isomorphism sending $i \in \mathbb{Z}_q$ to $iP$, using some version of the repeated squaring algorithm to perform the scalar multiplication in polynomial time. In a generic group, one assumes that instead of having explicit formulas for the group element $iP$, we have rather an "encording" $\sigma(i) \in S \subset \{0,1\}^*$, that represents the element $iP$. A generic algorithm $\mathcal{A}$ will then consult the oracle for two types of queries:

– $\mathcal{A}$ requests the encoding of $i$: the oracle will select randomly a value $\sigma(i)$, to represent the element $iP$, from the given set of bit-srings.
– Given two encodings $\sigma(i)$ and $\sigma(j)$, $\mathcal{A}$ requests (without knowing necessarily $i$ and $j$) the encoding of $(\sigma(i \pm j)$. Again the oracle responds with a randomly chosen bit-string.

The only condition on the oracle responses is that if the same group element is queried a second time, the same corresponding encoding must be returned. Without loss of generality, one could group the above queries in a single type of query, namely, $(\overrightarrow{i}, \overrightarrow{\alpha})$ where $\overrightarrow{i}$ refers to the set of indices of the group elements whereas $\overrightarrow{\alpha}$ denotes the set of exponents. In 5.2, we will give an interpretation of the oracle's behaviour regarding such a type of queries using polynomials over $\mathbb{F}_q$.

Finally, a security proof in this model assures the absence of an adversary who behaves generally with respect to the given group. However, and as in the random oracle model [7], a security proof in the generic model does not rule out the existence of a successful adversary for a specific group [9, 23].

**Reduction functions.** A reduction or conversion function is a map that sends an element of the group $G$ to an integer modulo the group size $q$. In DSA, the reduction function takes the group element, which is an integer modulo $p$, and reduces it modulo $q$. In ECDSA, the reduction function consists in taking the group element, which is a point $(x, y)$ in the given elliptic curve (over $\mathbb{F}_p$, where $p$ is not necessarily prime), interpret the $x$-coordinate as an integer in $\mathbb{Z}_p$, then outputs $x \bmod q$.

A reduction function must satisfy the so called *almost-invertibility*: given an arbitrary integer in $\mathbb{Z}_q$, then with nonnegligeable probability one can efficiently find one preimage. The almost-invertibility property allows to transfer the intractability of the discrete log problem on a generic group to $\mathbb{Z}_q$. ECDSA's reduction function satisfies this condition oppositely to DSA's function.

**Definition 6.** *Let $F$ be a reduction function $f : S \to \mathbb{Z}_q$. An almost-inverse of $F$ is a probabilistic algorithm $g$, possibly outputting $\perp$, such that:* $\Pr_{b \in_R \mathbb{Z}_q}[g(b) \in S \wedge F(g(b)) = b] \geq \frac{1}{3}$.

*Function $F$ is $(\delta, t)$-almost-invertible, with almost-inverse $g$, if furthermore:*

$$\mathcal{D} \approx_\delta \mathcal{U} \text{ where } \mathcal{D}_g = \{g(b) \mid b \in_R \mathbb{Z}_q \wedge g(b) \in \mathcal{G}\} \text{ and } \mathcal{U} = \{a \mid a \in_R \mathcal{G}\}$$

*The notation $\mathcal{D} \approx_\delta \mathcal{U}$ means that no distinguisher with running time $t$ can get an advantage greater than $\delta$.*

### 5.2 Preliminaries and notations

The EF-CMA-adversary is denoted $\mathcal{A}$ and will output after a certain number of queries to the group and the signing oracles ($n$ and $m$ respectively) a valid signature $\sigma = (R, T, s)$ on a message $(m, p)$ with success probability $\varepsilon = Succ_{\Sigma,\mathcal{A}}^{ef-cma}$. This event, is divided into subevents according to whether $R$ and $T$ were created during a signature or group query: a group element created during a group query will have a "grp" tag whereas tag "sign" will correspond to elements created in a signature query. Also, a signature query on $(m_i, p_i)$ will be always answered $(R_i, T_i, s_i)$, where $R_i, T_i \in S$. Hence we need to specify the type of an element that has the tag "sign": we denote type$(R) = 0$ and type$(T) = 1$.

The reduction, or more specifically the adversaries, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ against *random affine preimage*, *random linear collision* and *preimage* respectively will simulate the group and signing oracles according to the alleged kind of forgery returned by $\mathcal{A}$.

**The group oracle** As said previousely, the group oracle will receive queries of the type ( $\overrightarrow{i}$ , $\overrightarrow{\alpha}$) . The answers to such queries are elements $z_i$ of $S \subset \{0.1\}^*$. Let $\mathcal{L} = \{z_1, z_2, z_3, \ldots, z_{n+2}\}$ be the sequence of queries' answers where $n$ denotes the total number of queries to the group oracle. We use an interpretation similar to the one in [23], using polynomials $F_i(X)$ over $\mathbb{F}_q$:

- Polynomials $F_1$ and $F_2$ are set to $F_1 = 1$ and $F_2 = X$, which correspond to the generator and the public key respectively. The corresponding bit-strings are $z_1$ and $Z_2$ respectively.
- At the $\ell$-th query ( $\overrightarrow{i}$ , $\overrightarrow{\alpha}$), the polynomial $F_\ell$ is defined as $\sum_{j=1}^{|\overrightarrow{\alpha}|} \alpha_j \mathcal{F}_{\overrightarrow{i}_j}$. If $F_\ell$ is already listed[7] as $F_h$, then $F_\ell$ is marked and the corresponding answer to $F_h$ is returned. Otherwise, $z_\ell$ is selected at random from $S$, recorded using `Record` [8] ($z_\ell \| F_\ell \| \text{grp} \| \text{undef}$) and then returned to $\mathcal{A}$.

It is easy to see that the behavior driven by this interpretation is similar to the one of the regular algorithm provided that all the answers corresponding to unmarked polynomials are distinct and no polynomial $F_\ell$ vanishes at $x$. In these conditions, we call the sequence of encodings a safe sequence. The probability of such a sequence is given by the following lemma [23]:

**Lemma 1.** *Assume $n^2 \leq q$. The probability of unsafe sequence is upper-bounded by $5(n+1)^2/q$.*

Which follows from the following lemma:

**Lemma 2.** *Let $P$ be a non-zero affine polynomial in $\mathbb{Z}_q[X]$, then $\Pr_{x \in \mathbb{Z}_q}[P(x) = 0] = \frac{1}{q}$*

**The signing oracle $\Sigma$** The signing oracle $\Sigma$ will receive queries, of the form $(m, p)$ and will respond then with a valid signature $\sigma = (R, T, s)$ according to the following simulation:

**Simulation of $\Sigma$**: on query $(m, p)$ do the following:

- $R \overset{R}{\longleftarrow} S$, $e_p \leftarrow H_1(p, v)$, $d \leftarrow H_2(m, e_p, R)$,
- `Repeat`: $a, b \overset{R}{\longleftarrow} \mathbb{Z}_q$, $t \leftarrow (a - b \cdot F(R))a^{-1}d^{-1}v^{-1}h(m)^{-1} \bmod q$ `Until` $T = g(t) \neq \perp$,
- `Record` ($R \| aX + b \| \text{sign} \| 0$), `Record`($T \| d \| \text{sign} \| 1$),
- $s \leftarrow (d \cdot v \cdot t \cdot h(m) - 1)b^{-1} \bmod q$,
- `Return` $(R, T, s)$.

[7] The reduction (the adversaries $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$) will maintain, in addition to the outputs' list $\mathcal{L}$, three further lists, namely, the list of corresponding polynomials, denoted $\mathcal{F}$, the list of tags $\mathcal{T}$ to specify whether the group elements were created during a query to the group or signing oracles and the list of types $\mathcal{S}$ specifying whether the element is an $R$ or a $T$.

[8] The command `Record` ($R \| F \| t \| s$) will abort in some cases, namely when $(R, \star, \star, \star)$ already exists and $\star \neq F$. The probability, taken over the random choices of $R$, $F$, $t$ and $s$, of such an event to happen can be upper-bounded by $\frac{n}{q}$, where $n$ is the number of queries to $\Gamma$ (Lemma 2).

**The confirming/denying oracles** The use of designated verifier proofs of membership and of the registered public key model makes these oracles useless for the attacker. Therefore, we do not describe them in our security proof.

### 5.3 Unforgeability proof

As mentioned before, the EF-CMA-adversary $\mathcal{A}$ will output after a certain number of queries to $\Gamma$ and $\Sigma$ a signature $(R, T, s)$ on a message $(m, p)$ with success probability $\varepsilon = Succ_{\Sigma,\mathcal{A}}^{ef-cma}$. This event could be divided into sub-events, whose probabilities sum up to $\varepsilon$, according to the tags (*sign* or *grp*) and types (0 or 1 ) associated to both $R$ and $T$.
The reduction (adversaries, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ against *random affine preimage*, *random linear collision* and *preimage* respectively) will generate four random coins $c_i \in \{0, 1\}, 1 \leq i \leq 4$, and then simulates $\Gamma$ and $\Sigma$ accordignly.
More precisely, adversary $\mathcal{C}$ will use the forgery to solve *random linear collision* if it is of the form $\text{tag}(R, T) = (\text{sign}, \text{sign}) \wedge \text{type}(R, T) = (0, 1)$, whereas, $\mathcal{D}$ will use exploit a forgery of the form $\text{tag}(R, T) = (\text{grp}, \text{grp})$ to solve *preimage*, finally, adversary $\mathcal{B}$ will tilize all the remaining cases to solve *random affine preimage.*

**Theorem 1.** *Given an EF-CMA-adversary $\mathcal{A}$, operating in time $t$, after $n$ group queries and $m$ signing queries, such that $m \ll n^2$ and $n \gg 1$, with success probability $\varepsilon$, then there exist adversaries $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ operating in time $t'$ and attempting to solve* random affine preimage, random linear collision *and* preimage *with success probability $\mathbf{Succ}_{h,\mathcal{B}}^{raPre(n)}$, $\mathbf{Succ}_{h,\mathcal{C}}^{rlColl(n)}$ and $\mathbf{Succ}_{h,\mathcal{D}}^{Pre(n)}$ respectively such that:*

$$t' \leq t + 5n\tau_g \ln n + 5m \ln n(2\tau_g + \tau_{H_1} + \tau_{H_2} + \tau_F + \tau_h)$$

*and*

$$6\mathbf{Succ}_{h,\mathcal{B}}^{raPre(n)} + 2\mathbf{Succ}_{h,\mathcal{C}}^{rlColl(n)} + 3n^2\mathbf{Succ}_{h,\mathcal{D}}^{Pre(n)} \geq \frac{\varepsilon}{8} - 5n^4/q - 3mn^3$$

*where $\delta$ is the advantage of an adversary playing a distinguisher for $g$, $\tau_g$, $\tau_F$, $\tau_{H_1}, \tau_{H_2}$ and $\tau_h$ are the running time for $g$, $F$, $H_1$, $H_2$ and $h$ respectively.*

*Proof.* Let $(R, T, s)$ be the forgery output by $\mathcal{A}$ on $(m, p)$. Due to space limitations, we will detail only the case $\text{tag}(R, T) = (\text{grp}, \text{sign}) \wedge \text{type}(R, T) = (0, 1)$ and give a sketch of the cases $\text{tag}(R, T) = (\text{sign}, \text{sign}) \wedge \text{type}(R, T) = (0, 1)$ and $\text{tag}(R, T) = (\text{grp}, \text{grp})$ .
**Adversary** $\mathcal{B}$ generates three random coins $c_i \in \{0, 1\}, 1 \leq i \leq 3$ if $c_1 = 0 \wedge c_2 = 1 \wedge c_3 = 1$. This case corresponds to the event $S_0 : \text{tag}(R, T) = (\text{grp}, \text{sign}) \wedge \text{type}(T) = 1$ whose success probability is $\Pr[S_0] = \epsilon_1$. Then, the forgery returned by $A$ satisfies the following equation[9] $a - b \cdot F(R) = (ad - bc) \cdot v \cdot F(T) \cdot h(m)$, where $R = aU + bP$ and $T = cU + dP$. Since $T$ was generated during a signature query as a $T$ ($\text{type}(T) = 1$) then $c = 0$ (the adversary must know the

[9] this follows from the verification equation $(v \cdot F(T) \cdot h(m))T = F(R) \cdot U + s \cdot R + P$

discrete logarithm of $T$ in base $P$ in case the attacker asks for the signature conversion), the equation turns out to be $a - b \cdot F(R) = a \cdot d \cdot v \cdot F(T) \cdot h(m)$ or $1 - \frac{a}{b} F(R) = d \cdot v \cdot v \cdot F(T) \cdot h(m)$. Thus, in order to solve (random affine preimage), $\mathcal{B}$ must plunge $\alpha$ ($\beta$) in the group (signature) queries' answers. More precisely, he must answer group queries $(a, b)$ by $R$ such that $1 - \frac{a}{b} F(R) = \alpha$, similarly, signature queries must be answered by $(R, T, s)$, such that $-d \cdot v \cdot v \cdot F(T) = \beta$:

- **Game 1.** We use the interpretation given in 5.2 which considers a safe sequence $\mathcal{L}$. This event' s probability $\Pr[S_1]$ satisfies $|\Pr[S_1] - \Pr[S_0]| \leq 5(n+1)^2/q$.
- **Game 2.** In this game we simulate $\Gamma$. On query $(a, b)$ such that the corresponding polynomial $F = aX + b$ is unmarked, do the following:
  - `Repeat:` `pick` $\alpha$ from the corresponding oracle - `compute` $r \leftarrow (1 - \alpha)ab^{-1}$ - `compute` $\tilde{R} \leftarrow g(r)$ Until $\tilde{R} \neq$ Fail. However, we stop after $5 \ln n$ trials. This event which we denote $S_{2,1}$ differs from the previous one if $\tilde{R}$ remains undefined. Since the experiments are mutually independent ($a$ and $b$ are uniformly distributed), we may use a lemma from elementary probability theory ([23],Lemma 5) to bound the corresponding probability by $1/n^2$. The overall probability when $l$ ranges the set of queries indices is then $1/n$. Hence $\Pr[S_{2,1}] \geq (1 - 1/n) \Pr[S_1]$.
  - `Replace` $\tilde{R}$ by $R$. Since the inputs to $g$ are uniformly distributed ($\alpha$ is picked at random), we can use $n$ times the *almost-invertibility* of $F$ (the so-called *Hybrid Technique*) to bound the probability of this event ( denoted $S_{2,2}$): $|\Pr[S_{2,2}] - \Pr[S_{2,1}]| \leq n\delta_g$.
  - `Record`$(R \| aX + b \| \text{grp})$. This event, which we denote $S_{2,3}$ differs from the previous one if the command `Record` fails. This won't happen because of the assumption that $\mathcal{L}$ is a safe sequence. So $\Pr[S_{2,3}] = \Pr[S_{2,2}]$
  - `Return` $R$
- **Game 3.** In this game, we simulate the signing oracle. On query $(m, p)$ do the following:
  - Event $S_{3,1}$: `Compute` $e_p \leftarrow H_1(p, v)$ - `Pick` from the corresponding oracle - `Compute` $d \leftarrow H_2(m, e_p, R)$. Again, this game does not differ from the previous one, thus $\Pr[S_{3,1}] = \Pr[S_{2,3}]$
  - Event $S_{3,2}$: `Repeat:` pick $\beta$ from the corresponding oracle - compute $t \leftarrow -d^{-1}v^{-1}\beta$ `Until` $\tilde{T} = g(t) \neq$ Fail. However, we abort after $5 \ln n$ trials. Following the same argument above, we have $\Pr[S_{3,2}] \geq (1 - m/n^2) \Pr[S_{3,1}]$.
  - Event $S_{3,3}$: `Replace` $\tilde{T}$ by $T$. Again, applying the same technique, we get $|\Pr[S_{3,3}] - \Pr[S_{3,2}]| \leq m\delta_g$.
  - Event $S_{3,4}$: `Record`$(T \| d \| \text{sign} \| 1)$. This game differs from the previous if the command `Record` fails, du to the randomness of $d$, we can bound this probability by $|\Pr[S_{3,4}] - \Pr[S_{3,3}]| \leq mn/q$
  - Event $S_{3,5}$: `Pick` $a \in \mathbb{Z}_q^*$, `compute` $b \leftarrow a(\beta h(m) - 1)F(R)^{-1}$. We clearly have have $\Pr[S_{3,5}] = \Pr[S_{3,4}]$
  - Event $S_{3,6}$`Record`$(R \| aX + b \| \text{sign} \| 0)$, again, the diffence betwen the previous event is when `Record` fails. So $|\Pr[S_{3,6}] - \Pr[S_{3,5}]| \leq mn/q$.

  - Event $S_{3,7}$ `compute` $s \leftarrow -F(R)a^{-1}$. Due to the randomness of $a$, $\Pr[S_{3,7}] = \Pr[S_{3,6}]$.
  - Return $(R, T, s)$

- **Game 4.** In this game, $\mathcal{B}$ exploits the forgery $(R, T, s)$ returned by $\mathcal{A}$. If $\mathrm{tag}(R, T) = (\mathrm{grp}, \mathrm{sign}) \wedge \mathrm{type}(R, T) = (0, 1)$ and $\mathcal{B}$ generated the correct bits, then according to the above simulation there exits $i, j$ such that $R = R_i, T = T_j$ and $1 - \frac{a_i}{b_i}F(R_i) = \alpha_i$ and $-d_j \cdot v \cdot F(T_j) = \beta_j$, the equation satisfied by the forgery turns out to be $\alpha_i + \beta_j h(m) = 0$. $\mathcal{B}$ would then solve *random affine preimage* with success probability $Adv\mathcal{B} \geq \epsilon_1/8 + 5n^2/q - n\delta - m\delta - 2mn/q$ and time $t' \leq t + 5n \ln n + m(\tau_{H_1} + \tau_{H_2} + 5\tau_g \ln n + \tau_h + 2\tau_F)$.

**Adversary** $\mathcal{C}$ generates four random bits $c_i \in \{0, 1\}$. If $(c_1, c_2, c_3, c_4) = (1, 1, 0, 1)$, then $\mathcal{C}$ will simulate $\Gamma$ and $\Sigma$ such that the simulation exploits a forgery $(R, T, s)$ of the type $\mathrm{tag}(R, T) = (\mathrm{sign}, \mathrm{sign}) \wedge \mathrm{type}(R, T) = (0, 1))$. Hence $\mathcal{C}$ will simulate $\Gamma$ in the standard way described in 5.2. Furthermore, he will have to plunge the $\lambda$'s in answers to signature queries in a way that the returned signature $(R, T, s)$ satisfies $1 - \frac{b}{a}F(R) = \lambda$. More precisely, on $(m, p)$ $\mathcal{C}$ does the following: `Pick` $\lambda$ `- Compute` $e_p = H_1(p, v)$`- Repeat: pick` $\alpha \in_R \mathbb{Z}_q$, `compute` $r \leftarrow \frac{\lambda - 1}{\alpha}$ `Until` $R = g(r) \neq Fail$ `- Compute` $d = H_2(m, R, e_p)$ `- Repeat: pick` $a \in_R \mathbb{Z}_q$, `compute` $b = \alpha a$, `compute` $t = (a - bF(R))(a \cdot d \cdot v \cdot h(m))^{-1}$ `Until` $T = g(t) \neq Fail$ `- Record` $(R||aX + b||sign, 0)$ `- Record` $(T||d||sign||1)$ `- Compute` $s = (d \cdot v \cdot h(m) \cdot F(T) - 1) \cdot b^{-1}$ `- Return` $(R, T, s)$.
It is easy to conclude that this simulation, together with the above forgery returned by the attacker will lead to a solution to *random linear collision.*

**Adversary** $\mathcal{D}$ will attempt to exploit a forgery $(R, T, s)$ such that $\mathrm{tag}(R, T) = (\mathrm{grp}, \mathrm{grp})$ to find a preimage of a certain value, say $a$. The equation satisfied by the forgery is $a_i - b_iF(R_i) = (a_ib_j - a_jb_i)F(R_j) \cdot v \cdot h(m)$. For this, $\mathcal{D}$ will simulate the signing oracle in the standard way given in 5.2. To simulate $\Gamma$, $\mathcal{D}$ selects in advance $i, j \in_R [\![1, n]\!]$. If $i < j$, then on the $i$-th query $(a_i, b_i)$, $\mathcal{D}$ will select $R_i \in_R S$ and record it using `Record`$(R_i \| a_iX + b_i \| \mathrm{grp})$. On the $j-$th query $(a_j, b_j)$, `compute` $T_j \leftarrow g(a \cdot (a_i - b_iF(R))(a_ib_j - a_jb_i)^{-1}v^{-1})$. With probability at least $1/n^2$, $\mathcal{D}$ would have chosen the correct $i, j$ and the success of having $T_j \neq \perp$ is at least $1/3$ (almost invertibility of $F$ and randomness of $a$). If $j \leq i$, $\mathcal{D}$ will proceed in a similar manner.

### 5.4 Anonymity

**Theorem 2.** *Given an Ano-CMA-adversary $\mathcal{A}$, operating in time $t$, after $n$ group queries and $m$ signing queries, with success advantage $\varepsilon$, such that $m \ll n^2$, $m \ll q$ and $n \gg 2$, then there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{C}$, operating in time $t'$ and attempting to break the pseudo-randomness property of $H_1$, the pseudo-randomness of $H_2$ and the* random linear collision *of $h$ (respectively) with success probability $\mathbf{Succ}^{prf}_{H_1,\mathcal{B}_1}$, $\mathbf{Succ}^{prf}_{H_2,\mathcal{B}_2}$ and $\mathbf{Succ}^{rlColl(n)}_{h,\mathcal{C}}$ such that:*

$$t' \leq t + 5n\tau_g \ln n + 5m \ln n(\tau_{H_2} + \tau_h + \tau_g) + m\tau_{H_1}$$

*and*

$$\mathbf{Succ}^{prf}_{H_1,\mathcal{B}_1} + \mathbf{Succ}^{prf}_{H_2,\mathcal{B}_2} + 2\frac{\mathbf{Succ}^{rlColl(n)}_{h,\mathcal{C}}}{n} \geq \frac{\varepsilon}{n} + \frac{18n}{q} - n\delta + \delta + \frac{3m\delta}{n}$$

*where $\delta$ is the advantage of an adversary playing a distinguisher for $g$, $\tau_g$, $\tau_F$, $\tau_{H_1}, \tau_{H_2}$ and $\tau_h$ are the running time for $g$, $F$, $H_1$, $H_2$ and $h$ respectively.*

## 6 Discussion

We properly defined security notions for convertible undeniable signatures that support the additional property of *achronous* gradual conversion. Adapting the scheme proposed by Michels, Petersen and Horster in 1996, we realized the first scheme featuring this usefull notion of conversion. Moreover, we gave the first security analysis of the Michels-Petersen-Horster protocol, thereby addressing a problem left open since 1996. We have modified this scheme such that it becomes a generic one, which allows to use it for instance in the setting of elliptic curves (and therefore offers attractive practical advantages in terms of signature length and performances). In this context, in comparison with the only previous time-selective convertible undeniable signatures from [14], the computational costs for the confirmation/disavowal protocols and the conversion algorithms, are much smaller. We have proven the security of our scheme in the generic group model under new computational assumptions on the underlying hash functions.

## References

1. M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.*, Proceedings of the First ACM Conference on Computer and Communications Security (D. Denning, R. Pyle, R. Ganesan, R. Sandhu, and V. Ashby, eds.), ACM Press, 1993, pp. 62–73.
2. ———, *The Exact Security of Digital Signatures: How to Sign with RSA and Rabin.*, in Maurer [15], pp. 399–416.
3. J. Boyar, D. Chaum, I. B. Damgård, and T. B. Pedersen, *Convertible undeniable signatures.*, Advances in Cryptology - Crypto'90 (A. J. Menezes and S. A. Vanstone, eds.), Lect. Notes Comput. Sci., vol. 537, Springer, 1991, pp. 189–205.
4. G. Brassard, D. Chaum, and C. Crépeau, *Minimum disclosure proofs of knowledge.*, J. Comput. Syst. Sci. **37** (1988), no. 2, 156–189.
5. D. R. L. Brown, *Generic Groups, Collision Resistance, and ECDSA.*, Des. Codes Cryptography **35** (2005), no. 1, 119–152.
6. J. Camenisch and V. Shoup, *Practical Verifiable Encryption and Decryption of Discrete Logarithms.*, Advances in Cryptology - Crypto 2003 (D. Boneh, ed.), Lect. Notes Comput. Sci., vol. 2729, Springer, 2003, pp. 126–144.
7. R. Canetti, O. Goldreich, and S. Halevi, *The Random Oracle Methodology, Revisited.*, J. Assoc. Comput. Mach. **51** (2004), no. 4, 557–594.
8. D. Chaum and H. van Antwerpen, *Undeniable Signatures.*, Advances in Cryptology - Crypto'89 (G. Brassard, ed.), Lect. Notes Comput. Sci., vol. 435, Springer, 1990, pp. 212–216.

9. A. W. Dent, *Adapting the Weaknesses of the Random Oracle Model to the Generic Group Model.*, Advances in Cryptology - Asiacrypt 2002 (Y. Zheng, ed.), Lect. Notes Comput. Sci., vol. 2501, Springer, 2002, pp. 100–109.
10. S. D. Galbraith and W. Mao, *Invisibility and Anonymity of Undeniable and Confirmer Signatures.*, Topics in Cryptology - CT-RSA 2003 (M. Joye, ed.), Lect. Notes Comput. Sci., vol. 2612, Springer, 2003, pp. 80–97.
11. S. Goldwasser, S. Micali, and R. L. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks.*, SIAM J. Comput. **17** (1988), no. 2, 281–308.
12. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated Verifier Proofs and Their Applications.*, in Maurer [15], pp. 143–154.
13. C. Kudla and K. G. Paterson, *Non-interactive Designated Verifier Proofs and Undeniable Signatures.*, Cryptography and Coding, 10th IMA International Conference (N. P. Smart, ed.), Lect. Notes Comput. Sci., vol. 3796, Springer, 2005, pp. 136–154.
14. F. Laguillaumie and D. Vergnaud, *Time-Selective Convertible Undeniable Signatures.*, Topics in Cryptology - CT-RSA 2005 (A. J. Menezes, ed.), Lect. Notes Comput. Sci., vol. 3376, Springer, 2005, pp. 154–171.
15. U. M. Maurer (ed.), *Advances in Cryptology* - Eurocrypt *'96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, Lect. Notes Comput. Sci., vol. 1070, Springer, 1996.
16. M. Michels, H. Petersen, and P. Horster, *Breaking and Repairing a Convertible Undeniable Signature Scheme.*, Proceedings of the Third ACM Conference on Computer and Communications Security (L. Gong and J. Stern, eds.), ACM Press, 1996, pp. 148–152.
17. V. I. Nechaev, *Complexity of a Determinate Algorithm for the Discrete Logarithm.*, Math. Notes **55** (1994), no. 2, 165–172.
18. W. Ogata, K. Kurosawa, and S.-H. Heng, *The Security of the FDH Variant of Chaum's Undeniable Signature Scheme*, IEEE Trans. Inf. Theory **52** (2006), no. 5, 2006 – 2017.
19. T. Okamoto and D. Pointcheval, *The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes.*, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001 (K. Kim, ed.), Lect. Notes Comput. Sci., vol. 1992, Springer, 2001, pp. 104–118.
20. P. Paillier and D. Vergnaud, *Discrete-Log Based Signatures May Not Be Equivalent to Discrete-Log.*, Advances in Cryptology - Asiacrypt 2005 (B. Roy, ed.), Lect. Notes Comput. Sci., vol. 3788, Springer, 2005, pp. 1–20.
21. P. Rogaway and T. Shrimpton, *Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance.*, Fast Software Encryption, 11th International Workshop, FSE 2004 (B. K. Roy and W. Meier, eds.), Lect. Notes Comput. Sci., vol. 3017, Springer, 2004, pp. 371–388.
22. V. Shoup, *Lower Bounds for Discrete Logarithms and Related Problems.*, Advances in Cryptology - Eurocrypt'97 (W. Fumy, ed.), Lect. Notes Comput. Sci., vol. 1233, Springer, 1997, pp. 256–266.
23. J. Stern, D. Pointcheval, J. Malone-Lee, and N. P. Smart, *Flaws in applying proof methodologies to signature schemes.*, Advances in Cryptology - Crypto 2002 (M. Yung, ed.), Lect. Notes Comput. Sci., vol. 2442, Springer, 2002, pp. 93–110.