Time-Selective Convertible Undeniable Signatures with Short Conversion Receipts

Fabien Laguillaumie, Damien Vergnaud

Time-Selective Convertible Undeniable Signatures with *Short* Conversion Receipts *

Fabien Laguillaumie[1] Damien Vergnaud[2]

[1] GREYC, Université de Caen
Campus 2, Boulevard du Maréchal Juin, BP 5186
14032 Caen Cedex, France

[2] École normale supérieure – C.N.R.S. – I.N.R.I.A.
Département d'informatique, 45 rue d'Ulm
75230 Paris CEDEX 05, France

Abstract

Undeniable signatures were introduced in 1989 by Chaum and van Antwerpen to limit the self-authenticating property of digital signatures. An extended concept – the convertible undeniable signatures – proposed by Boyar, Chaum, Damgård and Pedersen in 1991, allows the signer to convert undeniable signatures to ordinary digital signatures.

In this article, we present a new efficient convertible undeniable signature scheme based on bilinear maps. Its unforgeability is tightly related, in the random oracle model, to the computational Diffie-Hellman problem and its anonymity to a non-standard decisional assumption. The advantages of our scheme are the short length of the signatures, the low computational cost of the signature and the receipt generation. Moreover, a variant of our scheme permits the signer to universally convert signatures pertaining only to a specific time period. We formalize this new notion as the *time-selective conversion*. We also improve our original scheme from CT-RSA'05 by reducing the length of the generated receipts: their size is now *logarithmic* in the number of time periods.

Keywords. Convertible undeniable signatures; bilinear maps; anonymity; exact security; time-selective conversion.

1 Introduction

We present a new efficient convertible undeniable signature scheme based on bilinear maps. Its unforgeability is tightly related, in the random oracle model, to the computational Diffie-Hellman problem and its anonymity to a non-standard decisional assumption. The advantages of our scheme are the short length of the signatures, the low computational cost of the signature and the receipt generation. Moreover, a variant of our scheme permits the signer to universally convert signatures pertaining only to a specific time period. We formalize this new notion as the *time-selective conversion*. Contrary to the original scheme [24] where the size of the receipts grew linearly with the number of time periods, the size of the generated receipts in the scheme proposed in this longer version is only *logarithmic* in the number of time periods.

1.1 Previous work

Digital signatures aim at recover *in silico* the usual properties of the traditional *in vivo* signatures, namely authentication, integrity, non-repudiation of the signed document and universal verifiability of the signatures. However, unlike handwritten signatures, digital signatures can be copy-cloned and

*This is the full version of "Time-Selective Convertible Undeniable Signatures" [24] presented at CT-RSA'05 by the same authors, with augmented results.

therefore authenticated confidential documents (*e.g.* software certificates, contracts, dishonourable bills) can easily be disseminated.

For privacy reasons, it is desirable, in many applications, that the verification of signatures be controlled or (at least) limited by the signer. This remark justifies the concept of undeniable signatures, introduced in 1989 by Chaum and van Antwerpen [13]. In this setting, the verification of a signature requires the cooperation of the signer. The security of their protocol relies on the hardness of the computational Diffie-Hellman problem, but suffers from the fact that the interactive protocols are not zero-knowledge. One year later, Chaum improved significantly the initial proposal in [11] by providing a zero-knowledge version.

There exist documents whose authentication must be limited at first, but which will require ordinary digital signatures after some period of time. 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*, introduced in [6] by Boyar, Chaum, Damgård and Pedersen, provide individual and universal conversions of the signatures. Unfortunately, the seminal Elgamal-like scheme has been broken in 1996 by Michels, Petersen, and Horster [26] who proposed a repaired version with heuristic security[1]. Since then, many schemes have then been proposed, based upon classical signatures, such as Schnorr [27], Elgamal [14] and RSA [18, 17, 16]. Convertible undeniable signatures have given rise to many applications in cryptography [6, 7, 12].

In all these protocols, the universal conversion 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 and furthermore the corresponding keys cannot be used to generate undeniable signatures any more. As in a classical public key infrastructure the public key needs to be certified by an authority (as well as in any asymmetric cryptographic protocol), this approach leads to the registration of a large number of public/secret key pairs for the signer, and the need for the verifiers to check the validity of these certificates.

1.2 Our contributions

In this article, we propose a new convertible undeniable signature scheme, in the spirit of both the original paper of Chaum and van Antwerpen [13] and the short signatures from bilinear maps proposed by Boneh, Lynn and Shacham [5]. The amusing fact is that the idea underlying these two papers is actually the same. In both cases, a signature consists in an exponentiation of (a hash-value of) the message by the signer's secret key : $h(m)^s$. In Chaum and van Antwerpen's scheme, the anonymity of signatures [16] comes from the difficulty of the decisional Diffie-Hellman problem in a prime order subgroup of the multiplicative group of a finite field, whereas the efficiency of Boneh *et al.* signatures comes from the ease of this problem on certain elliptic curves.

We combine the best of both worlds and, introducing Zhang–Safavi-Naini–Susilo and Boneh–Boyen's technique from [33, 3], we obtain a convertible undeniable signature protocol which is one of the most efficient. Moreover, to overcome the difficulty mentioned above, we introduce and formalize the *time-selective convertible undeniable signatures* which supports signers in gradually converting the undeniable signatures in a controlled fashion. A slight variant of our new scheme permits the signer to universally convert signatures pertaining only to a specific time period. The size of the receipt is *only* logarithmic in the number of time conversions (whereas in our previous proposal [24], this size was linear in the number of time periods).

The new convertible undeniable signature scheme is designed for devices with constrained computation capabilities or with low bandwidth. It can be embedded in smart cards for example, as the main computation for a signature is a scalar multiplication on an elliptic curve, and the signature is essentially one point (with some additional random salt). The unforgeability of our scheme is tightly related, in the random oracle model, to the computational Diffie-Hellman problem and its anonymity to a non-standard (non-interactive[2]) decisional assumption.

[1] The security of the scheme from [26] has recently been established in the generic group model [15].

[2] In [24], the assumption underlying the anonymity property was interactive.

1.3 Subsequent work

In [15], El Aimani and the second author provided a protocol for time-selective convertible undeniable signatures (based on [26]) with an advanced feature which permits the signer to universally convert *achronously* all signatures pertaining to a specific time period. The security is proven in the generic group model assuming specific assumptions on the hash function. Unfortunately, the conversion properties is ensured by using pseudo-random functions and are therefore not publicly checkable.

The technique introduced in [24] has had other applications. For instance, Paillier and the authors designed a very efficient universally convertible directed signatures [23] – such a construction had remained open since 1993.

1.4 Roadmap

The article is organised as follows: first we formally define the concept of time-selective convertible undeniable signature scheme and its security model in Section 2. Then, we recall in Section 3 standard techniques to (dis)prove the membership of group elements to some algebraic languages, we review the cryptographic properties of bilinear maps and we describe the problems upon which depend our schemes. In section 4, we describe our new scheme and its time-selective convertible variant. Eventually in Section 5 we prove their security in the random oracle model. In appendix, we present a time-selective convertible variant of Huang, Mu, Susilo and Wei's undeniable signature scheme from [20], since its security holds under more classical assumptions.

2 Formal definition and security model

2.1 Notations

The set of n-bit strings is denoted by $\{0,1\}^n$ and the set of all finite binary strings is denoted by $\{0,1\}^*$. The empty word is denoted by ϖ. 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 σ the probability that $\mathcal{A}$, on input x, outputs σ 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 $x \xleftarrow{R} S$. For a finite set X, $x \xleftarrow{R} X$ denotes a PPTM that samples a random element uniformly at random from X.

2.2 Definition

In this subsection, we formalise the concept of time-selective convertible undeniable signatures.

Definition 1 (Time-selective convertible undeniable signature)**.** *Let T be a positive integer. A* time-selective convertible undeniable signature scheme *with T time periods Σ is a 9-tuple*

$$\Sigma = (\mathsf{Setup}, \mathsf{SKeyGen}, \mathsf{VKeyGen}, \mathsf{Sign}, \mathsf{Control}, \mathsf{Confirm}, \mathsf{Deny}, \mathsf{Convert}, \mathsf{Verify})$$

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 $\mathcal{P}$*. The integer 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 $[\![1, T]\!]$ and a signing secret key as inputs and outputs a bit string.*

- $\Sigma.\mathsf{Control}$, *the* controlling algorithm, *is a PPTM which takes the public parameters, a message* m, *a bit string* σ, *an integer* $p \in [\![1, T]\!]$ *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* σ *is said to be a* signature *on* m *for* $\mathbf{pk_s}$ *for the time period* p.
- $\Sigma.\{\mathsf{Confirm}.\mathsf{Deny}\}$, *the* confirming/denying protocols *(respectively), are two-party protocols* $(\mathsf{Prove}, \mathsf{Verify})$ *such that:*
  - Prove *and* Verify *take as input the public parameters, a message* m, *an integer* $p \in [\![1, T]\!]$, *a bit-string* σ, *a signing public key* $\mathbf{pk_s}$ *and a verifying public key* $\mathbf{pk_v}$ *and the public parameters;*
  - Prove *takes as input* $\mathbf{sk_s}$ *the signing secret key corresponding to* $\mathbf{pk_s}$*;*
  - Verify *takes as input* $\mathbf{sk_v}$ *the verifying secret key corresponding to* $\mathbf{pk_v}$*;*

  $\mathsf{Confirm}.\mathsf{Verify}$ *(resp* $\mathsf{Deny}.\mathsf{Verify}$ *) outputs an element in* $\{\perp, 1\}$ *(resp* $\{\perp, 0\}$*).*
- $\Sigma.\mathsf{Convert}$, *the* conversion algorithm, *is a PPTM which takes as input the public parameters, an integer in* $[\![1, T]\!]$, *a signing key pair and a bit string* Υ *(either a pair message/signature or the empty string* ϖ*) and outputs a bit string.*
- $\Sigma.\mathsf{Verify}$, *the* verifying algorithm for converted signature, *is a PPTM which takes as input the public parameters, a message* m, *and a bit string* σ, *an integer* $p \in [\![1, T]\!]$, *a signing public key* $\mathbf{pk_s}$ *and a bit string* Λ *and outputs a bit. If the bit output is* 1 *then the bit string* Λ *is said to be a* receipt *of the validity of* σ.

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

$$\{(\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})) \in \Sigma.\mathsf{Setup}[k] \times \{0,1\}^{*^2} \times [\![1, T]\!] \times \Sigma.\mathsf{SKeyGen}[\mathcal{P}] \text{ such that } \Sigma.\mathsf{Control}[\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})] = \{1\}\}$$
$$\{(\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})) \in \Sigma.\mathsf{Setup}[k] \times \{0,1\}^{*^2} \times [\![1, T]\!] \times \Sigma.\mathsf{SKeyGen}[\mathcal{P}] \text{ such that } \Sigma.\mathsf{Control}[\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})] = \{0\}\}$$

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

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{sk_s}], \Sigma.\mathsf{Control}[\mathcal{P}, m, \sigma, p, \mathcal{S}] = \{1\}$$

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{sk_s}], \forall \Lambda \in \Sigma.\mathsf{Convert}[\mathcal{P}, p, \mathcal{S}, (m, \sigma)], \Sigma.\mathsf{Verify}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\}$$

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, p, \mathbf{pk_s}], \forall \Lambda \in \Sigma.\mathsf{Convert}[\mathcal{P}, p, \mathcal{S}, \varpi], \Sigma.\mathsf{Verify}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\}$$

$$\forall \sigma, \Lambda \in \{0,1\}^*, \Sigma.\mathsf{Verify}[\mathcal{P}, m, \sigma, p, \mathbf{pk_s}, \Lambda] = \{1\} \Rightarrow \Sigma.\mathsf{Control}[\mathcal{P}, m, \sigma, p, (\mathbf{sk_s}, \mathbf{pk_s})] = \{1\}.$$

Remark 1. *Informally, the fact that the protocols* $\Sigma.\mathsf{Confirm}$ *and* $\Sigma.\mathsf{Deny}$ *are designated verifier proof of membership systems implies the following security properties [22]:*

1. completeness 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. 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/invalidity of a signature.*

Our definition generalises the definition of traditional convertible undeniable signature since:

Definition 2 (Convertible Undeniable Signature)**.** *A* convertible undeniable signature scheme *is a time-selective convertible undeniable signature scheme with one time period.*

2.3 Security model

In this subsection, we define the quantitative notions of unforgeability and anonymity of a time-selective convertible undeniable signature scheme. The proofs of security are carried in the random oracle model proposed by Bellare and Rogaway [1]. In this model, hash functions are idealised as oracles which output a random value for each new query.

2.3.1 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].

2.3.2 Security against existential forgery under chosen message attack

The standard notion of security for digital signatures was defined by Goldwasser, Micali and Rivest [19] as *existential forgery against adaptive chosen message attacks* (EF-CMA). In [24], the corresponding notion for time-selective convertible undeniable signatures is defined along the same lines. The definition of *resistance to forgery* for time-selective convertible undeniable signatures that we propose is similar. In fact, we suppose that the adversary has access to the universal receipts for every time period $p \in [\![1, T]\!]$ and is allowed to query a converting oracle $\mathfrak{Cv}$, a confirming oracle $\mathfrak{C}$ and 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 3 (Unforgeability - EF-CMA)**.** *Let T be a positive integer, let*

$$\Sigma = (\mathsf{Setup}, \mathsf{SKeyGen}, \mathsf{VKeyGen}, \mathsf{Sign}, \mathsf{Control}, \mathsf{Confirm}, \mathsf{Deny}, \mathsf{Convert}, \mathsf{Verify})$$

be a time-selective convertible undeniable signature scheme with T time periods and let $\mathcal{A}$ be an PPTM. We consider the following random experiment, where k is a security parameter:

$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{ef-cma}}_{\Sigma,\mathcal{A}}(k)}$
$\mathcal{P} \xleftarrow{R} \Sigma.\mathsf{Setup}(k)$,
$(\mathbf{sk_s}, \mathbf{pk_s}) \xleftarrow{R} \Sigma.\mathsf{SKeyGen}(\mathcal{P})$
for $j = 1$ *to* T *do* $\Lambda_j \leftarrow \Sigma.\mathsf{Convert}(\mathcal{P}, j, (\mathbf{sk_s}, \mathbf{pk_s}), \varpi)$
$(m^\star, \sigma^\star, p^\star) \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{Cv},\mathfrak{C},\mathfrak{D}}(\mathcal{P}, \mathbf{pk_s}, \{\Lambda_j\}_{j\in[\![1,T]\!]})$
$\mathfrak{S} : (m, p) \longrightarrow \Sigma.\mathsf{Sign}(\mathcal{P}, m, p, \mathbf{sk_s})$
$\mathfrak{Cv} : (m, p, \sigma) \longrightarrow \Sigma.\mathsf{Convert}(\mathcal{P}, p, (\mathbf{sk_s}, \mathbf{pk_s}), (m, \sigma))$
$\mathfrak{C} : (m, p, \sigma, \mathbf{pk_v}) \longrightarrow \Sigma.\mathsf{Confirm}(\mathcal{P}, m, p, \sigma, \mathbf{pk_v}, \mathbf{pk_s})$
$\mathfrak{D} : (m, p, \sigma, \mathbf{pk_v}) \longrightarrow \Sigma.\mathsf{Deny}(\mathcal{P}, m, p, \sigma, \mathbf{pk_v}, \mathbf{pk_s})$
return 1 if and only if the following properties are satisfied:
- $\Sigma.\mathsf{Verify}[\mathcal{P}, \mathbf{pk_s}, m^\star, \sigma^\star, p^\star, \Lambda_{p^\star}] = \{1\}$
- $\sigma^\star$ *was not obtained from* $\mathfrak{S}$

We define the success *of $\mathcal{A}$,* via $\mathbf{Succ}^{\mathsf{ef\text{-}cma}}_{\Sigma,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{ef\text{-}cma}}_{\Sigma,\mathcal{A}}(k) = 1\right]$.
Given $(k, t) \in \mathbb{N}^2$ and $\varepsilon \in [0, 1]$, the scheme Σ is said to be (k, t, ε)-EF-CMA secure, if no EF-CMA-adversary $\mathcal{A}$ running in time t has $\mathbf{Succ}^{\mathsf{ef\text{-}cma}}_{\Sigma,\mathcal{A}}(k) \geq \varepsilon$.

The scheme Σ 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.

[3] It is often necessary to require the security of the schemes even if the adversary is the key registration centre. In this case, one must replace the proof of knowledge associated to the key registration by a zero-knowledge one [29].

2.3.3 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 [16]. 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 $\mathbf{pk}_{\mathbf{s}0}$ and $\mathbf{pk}_{\mathbf{s}1}$ 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 time-selective 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 time period $p \in [\![1, p^\star - 1]\!]$. 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 4 (Anonymity - Ano-CMA)**.** *Let T be a positive integer, let*

$$\Sigma = (\mathit{Setup}, \mathit{SKeyGen}, \mathit{VKeyGen}, \mathit{Sign}, \mathit{Control}, \mathit{Confirm}, \mathit{Deny}, \mathit{Convert}, \mathit{Verify})$$

be a time-selective convertible undeniable signature scheme with T time periods and let $\mathcal{A}$ be an PPTM. We consider the following random experiments, for $r \in \{0, 1\}$, where k is a security parameter:

Experiment $\mathbf{Exp}^{\mathit{ano\text{-}cma}-r}_{\Sigma,\mathcal{A}}(k)$

$\mathcal{P} \xleftarrow{R} \Sigma.\mathit{Setup}(k)$

$(\mathbf{sk}_{\mathbf{s}_0}, \mathbf{pk}_{\mathbf{s}_0}) \xleftarrow{R} \Sigma.\mathit{SKeyGen}(\mathcal{P})$

$(\mathbf{sk}_{\mathbf{s}_1}, \mathbf{pk}_{\mathbf{s}_1}) \xleftarrow{R} \Sigma.\mathit{SKeyGen}(\mathcal{P})$

$(m^\star, p^\star, \mathcal{I}) \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{Cv},\mathfrak{C},\mathfrak{D}}(\mathit{find}, \mathcal{P}, \mathbf{pk}_{\mathbf{s}_0}, \mathbf{pk}_{\mathbf{s}_1})$

- $\mathfrak{S} : (m, p, i) \longrightarrow \Sigma.\mathit{Sign}(\mathcal{P}, m, p, \mathbf{sk}_{\mathbf{s}_i})$
- $\mathfrak{Cv} : (m, p, \sigma, i) \longrightarrow \Sigma.\mathit{Convert}(\mathcal{P}, p, (\mathbf{sk}_{\mathbf{s}i}, \mathbf{pk}_{\mathbf{s}_i}), (m, \sigma))$
- $\mathfrak{C} : (m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, i) \longrightarrow \Sigma.\mathit{Confirm}(\mathcal{P}, m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, \mathbf{pk}_{\mathbf{s}_i})$
- $\mathfrak{D} : (m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, i) \longrightarrow \Sigma.\mathit{Deny}(\mathcal{P}, m, p, \sigma, \mathbf{pk}_{\mathbf{v}}, \mathbf{pk}_{\mathbf{s}_i})$

$\sigma^\star \xleftarrow{R} \Sigma.\mathit{Sign}(\mathcal{P}, m, \mathbf{sk}_{\mathbf{s}_r}, p^\star)$

for j *from* 1 *to* $p^\star - 1$ *do*

  $\Lambda^0_j \leftarrow \Sigma.\mathit{Convert}(\mathcal{P}, j, \mathbf{pk}_{\mathbf{s}0}, \mathbf{sk}_{\mathbf{s}0}, \varpi)$

  $\Lambda^1_j \xleftarrow{R} \Sigma.\mathit{Convert}(\mathcal{P}, j, \mathbf{pk}_{\mathbf{s}1}, \mathbf{sk}_{\mathbf{s}1}, \varpi)$

$d \leftarrow \mathcal{A}^{\mathfrak{S},\mathfrak{Cv},\mathfrak{C},\mathfrak{D}}(\mathit{guess}, \mathcal{I}, \{\Lambda^0_j, \Lambda^1_j\}_{j \in [\![1, p^\star - 1]\!]})$

Return d

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

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

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

The scheme Σ is said to be Ano-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 $(k, t(k), \varepsilon(k))$-Ano-CMA secure.

Remark 2. *To obtain the security results, the executions of the confirming-denying protocols must be simulated in the random oracle model. In our scheme, these protocols are achieved by interactive zero-knowledge proofs of equality/inequality of (root of) discrete logarithms. More precisely, we use designated-verifier proofs [21]. By definition of these proofs, the adversary gains no information other than the validity/invalidity of the signature from its interaction with the signer. As the adversary can obtain this conviction by querying the receipt generating oracle, there is no loss of generality to suppose that it does not have access to the confirming/denying oracles.*

3 Background

3.1 Proof of equality or inequality of two discrete logarithms

Let $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ be two groups of the same prime order q and let P and g be generators of $\mathbb{G}$ and $\mathbb{H}$ (respectively). 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 $(U_1, V_1, u_2, v_2) \in \mathbb{G}^2 \times \mathbb{H}^2$ is a Diffie-Hellman quadruple (or not), *i.e.* belongs to the set

$$\mathsf{EDL}(\mathbb{G}, \mathbb{H}) = \{(U_1, V_1, u_2, v_2) \in \mathbb{G}^2 \times \mathbb{H}^2, \log_{U_1}(V_1) = \log u_2(v_2)\},$$

(or to the set $\mathsf{IDL}(\mathbb{G}, \mathbb{H}) = (\mathbb{G}^2 \times \mathbb{H}^2) \setminus \mathsf{EDL}(\mathbb{G}, \mathbb{H})$).

We denote, as in [10], the zero-knowledge proof of **e**quality of the **d**iscrete **l**ogarithm (EDL) of $V_1 \in \mathbb{G}$ in base U_1 and the one of $v_2 \in \mathbb{H}$ in base u_2 by

$$PK(x : V_1 = xU_1 \wedge v_2 = u_2^x).$$

We use the notation

$$PK(x : V_1 = xU_1 \wedge v_2 \neq u_2^x).$$

for the proof of **i**nequality of the **d**iscrete **l**ogarithms (IDL). We refer the reader to Chaum and Pedersen's paper [12] for the proof of equality, and the one of Camenisch and Shoup [9] for the proof of inequality.

In the design of the time-selective convertible undeniable signature scheme, we also need zero-knowledge proofs of equality/inequality of a root of a discrete logarithm: given $p \in \mathbb{N}$,

$$PK(x : V_1 = xU_1 \wedge v_2 = {u_2}^{x^p}) \text{ and } PK(x : V_1 = xU_1 \wedge v_2 \neq {u_2}^{x^p}).$$

Techniques to design efficient protocol can be found in [10, 8].

To face *blackmailing* or man in the middle (or *mafia*) attacks against our undeniable signatures, we use interactive *designated verifier proofs*, as introduced in [21] by Jakobsson, Sako, and Impagliazzo, in Chaum's proofs of equality and inequality of discrete logarithm of [9]. The idea is to replace the generic commitment scheme by a *trapdoor commitment* [21] 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 procedure is used to force the users to know the secret-key corresponding to their public key). Such a proof is naturally non-transferable since the designated-verifier could have produced a given proof. The previously mentioned proofs of equality/inequality of a root of a discrete logarithm can be found in [32, p. 262].

Eventually, Fig. 1 and 2 represent respectively a designated verifier proof of equality of two discrete logarithms and its simulation. Fig. 3 and 4 represents respectively a designated verifier proof of inequality of two discrete logarithms and its simulation.

3.2 Bilinear maps

Admissible bilinear maps have allowed the opening up of new territories in cryptography, making possible the realisation of protocols that were previously unknown or impractical.

Definition 5 (Admissible bilinear map [4])**.** *Let $(\mathbb{G}, +)$ and $(\mathbb{H}, \times)$ be two groups of the same prime order q and let us denote by P a generator of $\mathbb{G}$. An* admissible bilinear map *is a map $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{H}$ satisfying the following properties:*

- *bilinear: $e(aQ, bR) = e(Q, R)^{ab}$ for all $(Q, R) \in \mathbb{G}^2$ and all $(a, b) \in \mathbb{Z}^2$;*
- *non-degenerate: $e(P, P) \neq 1$.*
- *computable: there exists a polynomial time algorithm to compute e.*

Definition 6 (prime-order-BDH-parameter-generator [4])**.**
A prime-order-BDH-parameter-generator *is a probabilistic algorithm which takes as input a security parameter k and outputs a 5-tuple $(q, P, \mathbb{G}, \mathbb{H}, e)$ where q is a prime with $2^k < q < 2^{k+1}$, $\mathbb{G}$ and $\mathbb{H}$ are groups of order q, P generates $\mathbb{G}$, and $e : \mathbb{G}^2 \longrightarrow \mathbb{H}$ is an admissible bilinear map.*

Usually $\mathbb{G}$ can be considered as a subgroup of points on a (hyper)elliptic curve over a finite field, $\mathbb{H}$ as a subgroup of the multiplicative group of a related finite field and e as the Weil or Tate pairing [4].

EDL.Prove		
Prover		Verifier
$(a, b, k) \xleftarrow{R} [\![1, q-1]\!]^3$		
$C_1 = kU_1$		
$c_2 = u_2^k$		
$C_3 = aU_1 + bY$	$\xrightarrow{C_1, c_2, C_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, q-1]\!]$
$c = k - x(r+b) \mod q$	$\xrightarrow{a, b, c}$	
		$C_1 \stackrel{?}{=} cU_1 + (r+b)V_1$
		$c_2 \stackrel{?}{=} u_2^c \cdot v_2^{r+b}$
		$C_3 \stackrel{?}{=} aU_1 + bY$

$$V_1 = xU_1 \wedge v_2 = u_2^x$$

Verifier: $(pk, sk) = (Y, y)$ with $Y = yU_1$

Figure 1: Designated verifier proof of membership in $\mathsf{EDL}(\mathbb{G}, \mathbb{H})$ – Prove protocol

3.3 The xyz-Decisional Diffie-Hellman problem

The unforgeability of our scheme is related to the classical Diffie-Hellman problem:

Computational Diffie-Hellman (CDH): Let $(\mathbb{G}, +)$ be a group of prime order q, and let a, b be in $[\![1, q-1]\!]$. Given $(P, aP, bP) \in \mathbb{G}^3$, compute abP.

The design of the new scheme is connected to the following decisional problem:

xyz-Decisional Diffie-Hellman problem[4] **(xyz-DDH):**
Let $(\mathbb{G}, +)$ be a group of prime order q, and let x, y and z be in $[\![1, q-1]\!]$. Given $(P, xP, yP, zP, Q) \in \mathbb{G}^5$, decide whether $Q = xyzP$.

At first glance, this may seem very similar to the classical decisional Diffie-Hellman (DDH) problem (in fact, the associated computational problem is equivalent to the CDH problem). The DDH assumption, underlying the security of many cryptographic protocols does not hold in the bilinear setting. Even if it is easier than the DBDH problem [4], the xyz-DDH problem seems intractable. Considering the xyz-DDH problem and assuming its difficulty (combined with the ease of the DDH problem), we are able to design cryptographic protocols achieving a trade-off between authenticity and privacy. The time-selective convertible undeniable signature scheme is based on the following observations:

- Assuming the computational intractability of the xyz-DDH problem, given $(P, xP, yP, zP, Q) \in \mathbb{G}^5$, no one can efficiently decide whether $Q = xyzP$.
- Everyone can be convinced by someone knowing either x, y or z that $Q = xyzP$. This can be done thanks to the proofs of equality of two discrete logarithms mentioned above, and the equalities
$$e(Q, P) = e(yP, zP)^x = e(xP, zP)^y = e(xP, yP)^z.$$
- After the publication of xyP, everyone can decide whether $Q = xyzP$.

Our new protocol is designed according to this idea but relies on a stronger assumption.

3.4 A new decisional problem

Recently, Boneh and Boyen [2] proposed an efficient digital signature scheme whose security (in the standard security model) relies on the so-called ℓ-Strong Diffie-Hellman problem (ℓ-SDH). This computational problem is slightly weaker than the ℓ-CAA problem[5] introduced by Mitsunari, Sakai, Kasahara

[4] a.k.a. the *Triffie-Hellman problem*
[5] collusion attack algorithm with ℓ traitors

EDL.Fake		
Prover		Verifier
$(c, d, k) \xleftarrow{R} [\![1, q-1]\!]^3$		
$C_1 = cU_1 + dV_1$		
$c_2 = u_2^c \cdot v_2^d$		
$C_3 = kU_1$	$\xrightarrow{C_1, c_2, C_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, q-1]\!]$
$b = d - r \mod q$		
$a = k - by \mod q$	$\xrightarrow{a, b, c}$	
		$C_1 \stackrel{?}{=} cU_1 + (r+b)V_1$
		$c_2 \stackrel{?}{=} u_2^c \cdot v_2^{r+b}$
		$C_3 \stackrel{?}{=} aU_1 + bY$

$$V_1 = xU_1 \wedge v_2 = u_2^x$$

Verifier: $(pk, sk) = (Y, y)$ with $Y = yU_1$

Figure 2: Designated verifier proof of membership in $\mathsf{EDL}(\mathbb{G}, \mathbb{H})$ – Fake protocol

in 2002 in relation with the security of a traitor tracing scheme [28]. Like the DDH problem, thanks to the bilinear map, the decisional problem associated to ℓ-CAA is easy. Therefore, we introduce a decisional variant of ℓ-CAA, similar to the xyz-DDH problem which runs in 3 stages:

(ℓ, T)-xyz Decisional CAA Problem ((ℓ, T)-xyz-DCAA): Let $(\mathbb{G}, +)$ be a group of prime order q, let x, y and z be in $[\![1, q-1]\!]$ and ℓ and T be in $\mathbb{N}$.

Input:	$\left[(x^i P)_{i \in [\![0, 2T-1]\!]}, yP, zP, h, \left(\frac{x^T y}{x + h_k} P, h_k\right)_{k \in [\![1, \ell]\!]}\right]$ in $\mathbb{G}^{2T+2} \times [\![1, q-1]\!] \times (\mathbb{G} \times [\![1, q-1]\!])^\ell$, with $h \notin \{h_1, \ldots, h_\ell\}$
Oracle:	for a request $t^\star \in [\![1, T]\!]$, the oracle answers $\left[(x^i yP)_{i \in [\![1, t^\star - 1]\!]}, Q\right] \in \mathbb{G}^{t^\star}$
Output:	decide whether $(x + h)Q = x^{t^\star} yzP$

The anonymity of our convertible undeniable signature scheme (*i.e.* with one time period) is related to the $(\ell, 1)$-xyz-DCAA problem. The link between this problem and the ℓ-CAA problem from [28] is analogous to the one between the xyz-DDH problem and the CDH problem. We want to stress that, though non-standard, these problems achieve generic security (as defined by Shoup [30]). The proof is similar to the one of the generic security of ℓ-SDH [2].

To conclude this section, we quantify the new algorithmic assumption.

Definition 7 ((ℓ, T)-xyz-DCAA assumption)**.** *Let* Gen *be a prime-order-BDH-parameter-generator and let ℓ and T be two integers. Let D be an adversary that takes on input $(q, P, \mathbb{G}, \mathbb{H}, e)$ a 5-tuple generated by* Gen *and*

$$\left[(X_i)_{i \in [\![1, 2T-1]\!]}, Y, Z, h, (R_j, h_j)_{j \in [\![1, \ell]\!]}\right] \in \mathbb{G}^{2T+1} \times [\![1, q-1]\!] \times (\mathbb{G} \times [\![1, q-1]\!])^\ell$$

and returns a bit. We consider the following random experiments, where k is a security parameter, for $r \in \{0, 1\}$:

IDL.Prove		
Prover		Verifier
$(a, b, k_0, k_1, k_2) \xleftarrow{R} [\![1, q-1]\!]^5$		
$c_0 = v_2^{k_0}/u_2^{xk_0}$		
$C_1 = k_1 U_1 - k_2 V_1$		
$c_2 = u_2^{k_1}/v_2^{k_2}$		
$C_3 = aU_1 + bY$	$\xrightarrow{c_0, C_1, c_2, C_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, q-1]\!]$
$c = k_1 - xk_0(r+b) \mod q$		
$d = k_2 - k_0(r+b) \mod q$	$\xrightarrow{a, b, c, d}$	
		$c_0 \stackrel{?}{\neq} 1_{\mathbb{H}}$
		$C_1 \stackrel{?}{=} cU_1 - dV_1$
		$c_2 \stackrel{?}{=} c_0 u_2^c / v_2^{r+b}$
		$C_3 \stackrel{?}{=} aU_1 + bY$

$$V_1 = xU_1 \wedge v_2 \neq u_2^x$$

Verifier: $(pk, sk) = (Y, y)$ with $Y = yU_1$

Figure 3: Designated verifier proof of membership in $\mathsf{IDL}(\mathbb{G}, \mathbb{H})$ – Prove protocol

Experiment $\mathbf{Exp}^{(\ell,\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}-\mathsf{r}}_{\mathsf{Gen},D}(k)$

$\mathcal{P} = (q, P, \mathbb{G}, \mathbb{H}, e) \xleftarrow{R} \mathsf{Gen}(k)$
$x \xleftarrow{R} [\![1, q-1]\!]$
For i *from* 1 *to* $2T-1$ *do* $X_i \leftarrow x^i P$
$(y, z) \xleftarrow{R} [\![1, q-1]\!]^2$, $Y \leftarrow yP$, $Z \leftarrow zP$
$h_1, \ldots, h_\ell, h \xleftarrow{R} [\![1, q-1]\!]$
For j *from* 1 *to* ℓ *do* $R_j = x^T y(x + h_j)^{-1} P$
$t^\star \leftarrow D(\mathcal{P}, (X_i)_{i \in [\![1,2T-1]\!]}, Y, Z, h, (R_j, h_j)_{j \in [\![1,\ell]\!]})$
For k *from* 1 *to* $t^\star - 1$ *do* $Y_k \leftarrow x^k yP$
If $r = 0$ *then* $Q \leftarrow x^T yz(x + h)^{-1} P$ *else* $Q \xleftarrow{R} \mathbb{G}$
$d \leftarrow D(\mathcal{P}, t^\star, (X_i)_{i \in [\![1,2T-1]\!]}, Y, Z, h, (R_j, h_j)_{j \in [\![1,\ell]\!]}, (Y_k)_{k \in [\![1,t^\star-1]\!]}, Q)$
Return d

We define the corresponding advantage *of* D *in solving the* $(\ell, T) - xyz$*-DCAA problem* via*:*

$$\mathbf{Adv}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}}_{\mathsf{Gen},D}(k) = \left| \Pr\left[\mathbf{Exp}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}\text{-}\mathsf{0}}_{\mathsf{Gen},D}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}\text{-}\mathsf{1}}_{\mathsf{Gen},D}(k) = 1\right] \right|.$$

Given $(k, t) \in \mathbb{N}^2$ *and* $\varepsilon \in [0, 1]$, Gen *is said to be* (k, t, ε)*-*(ℓ, T)*-xyz-DCAA-secure if no adversary* D *running in time* t *has advantage*

$$\mathbf{Adv}^{(\ell,\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}}_{\mathsf{Gen},D}(k) \geq \varepsilon.$$

The generator Gen *is said to be* (ℓ, T)*-xyz-DCAA-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* $(k, t(k), \varepsilon(k))$*-*$(\ell, T)$*-xyz-DCAA-secure.*

IDL.Fake		
Prover		Verifier
$(c, d, k_1, k_2) \xleftarrow{R} [\![1, q-1]\!]^4$		
$c_0 \xleftarrow{R} \mathbb{H} \setminus \{1_{\mathbb{H}}\}$		
$C_1 = cU_1 - dV_1$		
$c_2 = c_0 u_2^c / v_2^{k_1}$		
$C_3 = k_2 U_1$	$\xrightarrow{c_0, C_1, c_2, C_3}$	
	$\xleftarrow{r}$	$r \in_R [\![1, q-1]\!]$
$b = k_1 - r \mod q$		
$a = b - k_2 y \mod q$	$\xrightarrow{a, b, c, d}$	
		$c_0 \stackrel{?}{\neq} 1_{\mathbb{H}}$
		$C_1 \stackrel{?}{=} cU_1 - dV_1$
		$c_2 \stackrel{?}{=} c_0 u_2^c / v_2^{r+b}$
		$C_3 \stackrel{?}{=} aU_1 + bY$

$$V_1 = xU_1 \wedge v_2 \neq u_2^x$$

Verifier: $(pk, sk) = (Y, y)$ with $Y = yU_1$

Figure 4: Designated verifier proof of membership in $\mathsf{IDL}(\mathbb{G}, \mathbb{H})$ – Fake protocol

4 A new convertible time-selective undeniable signature scheme

4.1 The new convertible undeniable signature scheme: CUSBM

In this section, we describe the new convertible undeniable signature scheme CUSBM, based on bilinear map. It is designed as follows:

Setup and Key Generation

Setup: Let k be a security parameter, Gen be a prime-order-BDH-parameter-generator and $(q, P, \mathbb{G}, \mathbb{H}, e)$ some output of $\mathsf{Gen}(k)$. Let $f_r : \mathbb{N} \to \mathbb{N}$ be a function. We denote $n_r = f_r(k)$. Let $[\{0,1\}^* \times \{0,1\}^{n_r} \longrightarrow \mathbb{G}]$ be a hash function family, and H be a random member of this family. Let $[\{0,1\}^* \times \{0,1\}^{n_r} \longrightarrow [\![1, q-1]\!]]$ be a hash function family, and h be a random member of this family. The public parameters are $[(q, P, \mathbb{G}, \mathbb{H}, e), H, h]$

SKeyGen: Alice picks randomly two integers $a_1, a_2 \in [\![1, q-1]\!]$ and computes the points $P_1 = a_1 P$ and $P_2 = a_2 P$. Alice's public key is the pair (P_1, P_2) and her secret key is (a_1, a_2).

VKeyGen: the verifier Bob picks randomly an integer $y \in [\![1, q-1]\!]$ and computes the point $Y = yP$. Bob's public key is Y and his secret key is y.

Signing and controlling algorithm

Sign: Given a message $m \in \{0,1\}^*$, Alice picks at random $r \in \{0,1\}^{n_r}$ and computes the point

$$\sigma = \frac{a_1 a_2}{(a_2 + h(m||r))} H(m||r).$$

The convertible undeniable signature of the message m is (σ, r).

Control: Given a message m and a putative signature (σ, r), Alice checks that

$$e(\sigma, P_2 + h(m||r)P) = e(H(m||r), P_1)^{a_2}.$$

Confirmation / Denial protocols

Confirm: Given a message m and a signature (σ, r), Alice can confirm (σ, r) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P_2 + h(m||r)P) = e(H(m||r), P_1)^{a_2} \wedge P_2 = a_2 P).$$

Deny: Given a message m and an invalid signature (σ, r), Alice can deny (σ, r) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P_2 + h(m||r)P) \neq e(H(m||r), P_1)^{a_2} \wedge P_2 = a_2 P).$$

Receipt generation and verification

Convert

- On input ϖ, Alice publishes the point $I = a_1 a_2 P$.

- Given a message $m \in \{0,1\}^*$ and a putative signature (σ, r) on m, Alice checks the validity of the signature thanks to the Control algorithm and then computes the point $\tilde{\sigma} = a_2 H(m||r) \in \mathbb{G}$. The individual receipt with respect to σ is $\tilde{\sigma}$.

Verify:

- The validity of the universal receipt I is decided by verifying that

  $$e(P_1, P_2) = e(I, P).$$

  If it is valid, given a signature (σ, r) on a message $m \in \{0,1\}^*$ and I, everyone checks the validity of this signature by verifying that

  $$e(\sigma, P_2 + h(m||r)P) = e(H(m||r), I).$$

- Given a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m and a putative individual receipt $\tilde{\sigma}$ on (σ, r), the validity of the receipt is decided by checking whether

  $$e(\tilde{\sigma}, P) = e(P_2, H(m||r))$$

  or not. If $\tilde{\sigma}$ is valid, then the validity of (σ, r) is decided by checking whether

  $$e(\sigma, P_2 + h(m||r)P) = e(\tilde{\sigma}, P_1)$$

  or not.

Remark 3. *Note that the designated verifier interactive proofs used in the confirm/deny protocols can be made non-interactive using the Fiat-Shamir paradigm (at the expense of the use of another random oracle). In addition, the scheme will gain in efficiency since the overall communication complexity will be reduced.*

Remark 4 (Efficiency considerations)**.** *Comparing with previous convertible undeniable signature schemes, CUSBM has a number of advantages. The signature only consists in an element of $\mathbb{G}$ and some additional random salt. In practise, the size of an element of $\mathbb{G}$ can be reduced by a factor 2 with compression techniques and the random salt has size $n_r = 60$. Therefore, the size of the signature is only 231 bits. Furthermore, a receipt (individual and universal) is also an element of $\mathbb{G}$, and therefore has bit size 171. From an efficiency point of view, the signature generation and the individual and universal receipts generation algorithms require only one exponentiation as the most expensive operation. Unfortunately, it turns out that the signature verification is slightly more time consuming, as it requires 2 pairing evaluations.*

4.2 A time-selective convertible undeniable signature scheme: TSCUSBM

TSCUSBM is a time-selective convertible undeniable signature scheme which is a variant of CUSBM.

Setup and Key Generation

TSCUSBM.Setup = CUSBM.Setup

SKeyGen: Alice picks randomly two integers $a_1, a_2 \in [\![1, q-1]\!]$, and computes the points $P_1 = a_1 P$ and $P_2 = a_2 P$. She chooses a number of time periods $T \in \mathbb{N}$. Alice's public key is the pair (P_1, P_2, T) and her secret key is (a_1, a_2).
TSCUSBM.VKeyGen= CUSBM.VKeyGen

Signing and controlling algorithm

Sign: Given a message $m \in \{0,1\}^*$ and a time period t, Alice picks at random $r \in \{0,1\}^{n_r}$ and computes the point $\sigma = a_1 a_2^t (a_2 + h(m||r))^{-1} H(m||r)$. The signature of the message m is (σ, r, t).
Control: Given a message m and a putative signature (σ, r, t), Alice checks that

$$e(\sigma, P_2 + h(m||r)P) = e(H(m||r), P_1)^{a_2^t}.$$

Confirmation / Denial protocols

Confirm: Given a message m and a signature (σ, r, t), Alice can confirm (σ, r, t) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P_2 + h(m||r)P) = e(H(m||r), P_1)^{a_2^t} \wedge P_2 = a_2 P).$$

Deny: Given a message m and an invalid signature (σ, r, t), Alice can deny (σ, r, t) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P_2 + h(m||r)P) \neq e(H(m||r), P_1)^{a_2^t} \wedge P_2 = a_2 P).$$

Receipt generation and verification

Convert:

- On input ϖ, and an integer $t \in [\![1, T]\!]$, this protocol consists for Alice in publishing the t-tuple

$$I_t = (a_1 a_2 P, a_1 a_2^2 P, \ldots, a_1 a_2^t P) \in \mathbb{G}^t.$$

- Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m for the time period t, Alice checks the validity of the signature thanks to the Control algorithm and then computes the t-tuple

$$\tilde{\sigma} = (a_2 H(m||r), a_2^2 H(m||r), \ldots, a_2^t H(m||r)).$$

  The individual receipt with respect to σ is $\tilde{\sigma}$.

Verify

- Given an integer $t \in [\![1, T]\!]$, the validity of the universal receipt $I_t = (I^{(1)}, \ldots, I^{(t)})$ is decided by verifying that

$$e(P_1, P_2) = e(I^{(1)}, P) \text{ and } e(I^{(i-1)}, P_2) = e(I^{(i)}, P) \text{ for } i \in [\![2, t]\!].$$

  If it is valid, given a signature (σ, r) on a message $m \in \{0,1\}^*$ for a time period $t' \leq t$ and I_t, everyone checks the validity of this signature by verifying that

$$e(\sigma, P_2 + h(m||r)P) = e(H(m||r), I^{(t')}).$$

- Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m for the time period t and a putative individual receipt $\tilde{\sigma} = (\tilde{\sigma}^{(1)}, \ldots, \tilde{\sigma}^{(t)})$ on (σ, r), the validity of the receipt is decided by checking whether $e(\tilde{\sigma}^{(1)}, P) = e(P_2, H(m||r))$ and $e(\tilde{\sigma}^{(i)}, P) = e(P_2, \tilde{\sigma}^{(i-1)})$ for $i \in [\![2, t]\!]$ or not. If $\tilde{\sigma}$ is valid, then the validity of (σ, r) is decided by checking whether

$$e(\sigma, P_2 + h(m||r)P) = e(\tilde{\sigma}^{(t)}, P_1)$$

or not.

Remark 5. *In a sequential use of these signatures (i.e. signatures for the time period t are individually converted only after the publication of I_{t-1}), the verification processes can be considerably improved (as in the CUSBM scheme). Otherwise, the signer's operations are still efficient, but the verifier has more computations to perform. The scheme remains nevertheless reasonable.*

4.3 An improved scheme with logarithmic size of the receipts: TSCUSBM+

To reduce the size of the receipts, we adapt an idea of Bresson and Stern from [8], and therefore we reach a logarithmic size of the receipts. The technique is essentially similar to a "double-and-add" algorithm and is described in the following figures. The whole TSCUSBM+ scheme is the same as TSCUSBM but the two following procedures which must be modified using the ReceiptGen and ReceiptVerif of the Fig. 5 and described below:

TSCUSBM+.Convert:

- On input ϖ, and an integer $t \in [\![1, T]\!]$, this protocol consists for Alice running ReceiptGen(P_1, a_2, t).

- Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m for the time period t, Alice checks the validity of the signature thanks to the Control algorithm and then runs ReceiptGen$(H(m||r), a_2, t)$.

TSCUSBM+.Verify

- Given an integer $t \in [\![1, T]\!]$, the validity of the universal receipt $I_t = (I^{(1)}, \ldots, I^{(t)})$ is decided by running ReceiptVerif(P_1, P_2, t, I_t) and checking that

$$e(\sigma, P_2 + h(m||r)P) = e(H(m||r), I^{(t)}).$$

- Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m for the time period t and a putative individual receipt $\tilde{\sigma} = (\tilde{\sigma}^{(1)}, \ldots, \tilde{\sigma}^{(t)})$ on (σ, r), the validity of the receipt is decided by running ReceiptVerif$(H(m||r), P_2, t, \tilde{\sigma})$. If $\tilde{\sigma}$ is valid, then the validity of (σ, r) is decided by checking whether

$$e(\sigma, P_2 + h(m||r)P) = e(\tilde{\sigma}^{(t)}, P_1)$$

or not.

5 Security results

5.1 Unforgeability

The theorem below states that TSCUSBM is EF-CMA secure assuming the intractability of the CDH problem in the random oracle model (we replace the hash function H by random oracle $\mathcal{H}$; there is no need to do ideal assumptions on h).

Theorem 1 (Unforgeability of TSCUSBM)**.** *Let* Gen *be a prime-order-BDH-parameter-generator, let $f_r : \mathbb{N} \to \mathbb{N}$ and let $\mathcal{A}$ be an EF-CMA-adversary against TSCUSBM in the random oracle model, that produces an existential forgery with probability $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{TSCUSBM,\mathcal{A}}$, within time τ, making $q_{\mathcal{H}}$, q_Σ and*

ReceiptGen
Input $\mathcal{P}$, Q, v, t
Output $\vec{T}$
$[\lambda_{\ell-1}, \ldots, \lambda_0] \leftarrow \mathsf{BinaryExpansion}(t)$
$i \leftarrow \ell - 1$
$\alpha \leftarrow 1$
while $i \geq 0$ do
$\alpha \leftarrow \alpha^2 \mod q$
$R_i \leftarrow \alpha Q$
if $\lambda_i = 1$ then $\alpha \leftarrow \alpha \cdot v \mod q$
$S_i \leftarrow \alpha Q$
else $S_i \leftarrow \varepsilon$
$i \leftarrow i - 1$
$I_t \leftarrow [(R_{\ell-1}, S_{\ell-1}), \ldots, (R_0, S_0)]$

ReceiptVerif
Input $\mathcal{P}$, Q, V, t,
and $[(R_{\ell-1}, S_{\ell-1}), \ldots, (R_0, S_0)]$
Output b
$[\lambda_{\ell-1}, \ldots, \lambda_0] \leftarrow \mathsf{BinaryExpansion}(t)$
$i \leftarrow \ell - 1$; $S_0 \leftarrow Q$
$b \leftarrow \mathsf{Valid}$
while $i \geq 0$ do
$A \leftarrow e(R_i, Q)$
if $S_{i-1} = \varepsilon$ then $B \leftarrow e(R_{i-1}, R_{i-1})$
else $B \leftarrow e(S_{i-1}, S_{i-1})$
if $A \neq B$ then $b \leftarrow \mathsf{Invalid}$
if $\lambda_i = 1$ then
$A \leftarrow e(S_i, Q)$
$B \leftarrow e(R_i, S_0)$
if $A \neq B$ then $b \leftarrow \mathsf{Invalid}$

Figure 5: Description of ReceiptGen and ReceiptVerif

q_Υ queries to the random oracle, to the signing oracle and to the receipt generating oracle. Then there exist $\varepsilon' \in [0,1]$ and $\tau' \in \mathbb{N}$ verifying

$$\begin{cases} \varepsilon' \geq \varepsilon - \dfrac{q_\Sigma(q_\mathcal{H} + q_\Sigma)}{2^{n_r}} \\ \tau' \leq \tau + (q_\mathcal{H} + q_\Sigma + O(1))T_{Exp-\mathbb{G}} + (q_\Sigma + O(1))T_{Exp-q} \end{cases}$$

such that CDH can be solved with probability ε', within time τ', and where $T_{Exp-\mathbb{G}}$ (resp. T_{Exp-q}) denotes the time for an exponentiation in $\mathbb{G}$ (resp. modulo q) and $n_r = f_r(k)$.

Proof: The proof is very similar to Boneh, Lynn and Shacham's unforgeability proof in [5].

We define a sequence of games Game_0, ..., Game_3 starting from the actual EF-CMA adversary $\mathcal{A}$ and modify it step by step, until we reach a final game whose success probability has an upper bound related to solving CDH. All the games operate on the same underlying probability space: the public and private keys of the signature scheme, the coin tosses of the adversary $\mathcal{A}$ and the random oracle $\mathcal{H}$. Let k be a security parameter, let $(q, P, \mathbb{G}, \mathbb{H}, e)$ be a 5-tuple generated by $\mathsf{Gen}(k)$ and let (X, Y) be a random instance of the CDH problem.

Game_0 We consider an EF-CMA-adversary $\mathcal{A}$ with success $\mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{TSCUSBM},\mathcal{A}}$, within time τ. The key generation algorithm is run and produces a pair of keys (pk, sk, T). The universal receipt generation algorithm produces the information $(\mathcal{I}_1, \ldots, \mathcal{I}_T)$. The adversary $\mathcal{A}$ is fed with pk and $(\mathcal{I}_1, \ldots, \mathcal{I}_T)$ and, querying the random oracle $\mathcal{H}$, the signing oracle Σ, and the receipt generating oracle Υ, outputs a couple $(m^\star, (\sigma^\star, r^\star, t^\star))$.

We denote by $q_\mathcal{H}$, q_Σ and q_Υ the number of queries from the random oracle $\mathcal{H}$, from the signing oracle Σ and from the receipt generating oracle Υ. The only requirement is that the output signature $(\sigma^\star, r^\star, t^\star)$ has not been obtained from the signing oracle. For a signing query on a message m, we first ask an hash value of m with some additional random salt r and when the adversary outputs its forgery $(m^\star, (\sigma^\star, r^\star, t^\star))$, we ask a hash value of $(m^\star, r^\star)$. Therefore at most $q_\mathcal{H} + q_\Sigma + 1$ queries are asked to the hash oracle during this game.

In any Game_j, we denote by $\mathsf{Forge}_\mathsf{j}$ the event

$$\mathsf{TSCUSBM.UVerify}(m^\star, (\sigma^\star, r^\star, t^\star), pk, \mathcal{I}_T) = 1.$$

By definition, we have $\Pr[\mathsf{Forge}_0] = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{TSCUSBM},\mathcal{A}}$.

Game_1 In this game, we pick an element $a_2 \in [\![1, q-1]\!]$ at random, and we modify the simulation by replacing pk by $(X, a_2 P)$ and the T-tuple $(\mathcal{I}_1, \ldots, \mathcal{I}_T)$ by $(a_2 X, a_2^2 X, \ldots, a_2^T X)$. The distribution of pk is unchanged since (X, Y) is a random instance of the CDH problem and a_2 is picked at random. From now on, thanks to the knowledge of a_2, we can simulate the receipt generating oracle Υ. We have

$$\Pr[\mathsf{Forge}_1] = \Pr[\mathsf{Forge}_0].$$

Game_2 In this game, we simulate the random oracle $\mathcal{H}$. For any fresh query $(m, r) \in \{0,1\}^* \times \{0,1\}^{n_r}$ to the oracle $\mathcal{H}$, we pick at random $u \in [\![1, q-1]\!]$ and compute $Q = (a_2 + h(m||r))uY$. We store (m, r, u, Q) in the H-List and return Q as the answer to the oracle call. In the random oracle model, this game is clearly identical to the previous one. Hence, we get

$$\Pr[\mathsf{Forge}_2] = \Pr[\mathsf{Forge}_1].$$

Game_3 In this game, we simulate the signing oracle Σ: for any message m, whose signature is queried for the time period t, we pick at random two elements $r \in \{0,1\}^{n_r}$, $v \in [\![1, q-1]\!]$, and compute $\sigma = a_2^t v(a_2 + h(m||r))^{-1}X$. If the H-List includes a quadruple $(m, r, ?, ?)$ we abort the simulation, else we store (m, r, v, vP) in the H-List and output σ as the signature. As we abort with probability at most $(q_{\mathcal{H}} + q_\Sigma)2^{-n_r}$, summing up the inequalities for all signature queries, we have

$$|\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_2]| \leq q_\Sigma(q_{\mathcal{H}} + q_\Sigma)2^{-n_r}.$$

When the game Game_3 terminates, outputting a valid pair message/signature $(m^\star, (\sigma^\star, r^\star, t^\star))$, by definition of existential forgery, the H-List includes a quadruple $(m^\star, r^\star, u^\star, Q^\star)$ with $Q^\star = (a_2 + h(m||r))u^\star Y$. By the simulation, we have $e(\sigma^\star, P_2 + h(m||r)P) = e(Q^\star, a_2^{t^\star} P_1)$, and the point $R = (u^\star a_2^{t^\star})^{-1}\sigma^\star$ is a solution to our instance of the Computational Diffie-Hellman Problem.

This concludes the proof. □

Corollary 1 (Unforgeability of CUSBM and TSCUSBM+)**.** *Under the CDH assumption, the schemes CUSBM and TSCUSBM+ are EF-CMA secure in the random oracle model.*

Remark 6. *In order to shorten the public keys, we can modify the key generation of CUSBM as follows : the key generation algorithm produces a public/private pair of keys (aP, a). This variant of CUSBM (setting $a_1 = a_2 = a$) is unforgeable under the ℓ-CAA [28] assumption.*

5.2 Anonymity

The next theorem claims TSCUSBM's anonymity against a chosen message attack under the (ℓ, T)-xyz-DCAA assumption in the random oracle model (the hash functions h and H are replaced by random oracles $\hbar$ and $\mathcal{H}$):

Theorem 2 (Anonymity of TSCUSBM)**.** *Let Gen be a prime-order-BDH-parameter-generator, let $f_r : \mathbb{N} \to \mathbb{N}$ and let $\mathcal{A}$ be an Ano-CMA-adversary against TSCUSBM in the random oracle model, that breaks anonymity with advantage $\mathbf{Adv}^{\mathsf{ano-cma}}_{\mathit{TSCUSBM},\mathcal{A}}$, within time τ, making $q_{\mathcal{H}}$, $q_\hbar$, q_Σ and q_Υ queries to the random oracles, to the signing oracles and to the receipt generating oracles. Then there exist $T \in \mathbb{N}$, $\varepsilon' \in [0,1]$ and $\tau' \in \mathbb{N}$ verifying*

$$\begin{cases} \varepsilon' \geq \dfrac{\varepsilon}{2} - \dfrac{(q_\Sigma + 1)(q_{\mathcal{H}} + q_\hbar)}{2^{n_r - 1}} - \dfrac{1}{2^k} \\ \tau' \leq \tau + (q_{\mathcal{H}} + 2q_\Sigma + Tq_\Upsilon + O(1))T_{Exp-\mathbb{G}} + (q_\Sigma + Tq_\Upsilon + O(1))T_{Exp-q} \end{cases}$$

such that $(q_{\mathcal{H}}, T)$-xyz-DCAA can be solved with probability ε', within time τ', and where $T_{Exp-\mathbb{G}}$ (resp. T_{Exp-q}) denotes the time for an exponentiation in $\mathbb{G}$ (resp. modulo q) and $n_r = f_r(k)$.

Proof: Our method of proof is inspired by Shoup [31]: we define a sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_5$ starting from the actual Ano-CMA adversary $\mathcal{A}$ and modify it step by step, until we reach a final game whose success probability has an upper bound related to solving the $(q_{\mathcal{H}}, T)$-xyz-DCAA problem. All the games operate on the same underlying probability space: the public and private keys of the signature scheme, the coin tosses of $\mathcal{A}$ and the random oracles.

Let k be a security parameter, T and ℓ be two integers and let $(q, P, \mathbb{G}, \mathbb{H}, e)$ be a 5-tuple generated by $\mathsf{Gen}(k)$ and $\big[(X_i)_{i \in [\![1, 2T-1]\!]}, Y, Z, h), (R_j, h_j)_{j \in [\![1,\ell]\!]}\big]$ in $\mathbb{G}^{2T+1} \times [\![1, q-1]\!] \times (\mathbb{G} \times [\![1, q-1]\!])^\ell$ be a random instance of the $(q_{\mathcal{H}}, T)$-xyz-DCAA problem. We denote $X_0 = P$. We construct a simulation which solves this instance.

Game_0 We consider an Ano-CMA-adversary $\mathcal{A}$ with advantage $\mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{TSCUSBM},\mathcal{A}}(k)$, within time τ. The key generation algorithm is run twice and produces the following pairs of keys (sk_0, pk_0, T_0), and (sk_1, pk_1, T_1). The adversary $\mathcal{A}$ is fed with pk_0, pk_1, T_0 and T_1 and, querying the random oracles $\mathcal{H}$ and $\hbar$, the signing oracles Σ_0 and Σ_1, and the receipt generating oracles Υ_0 and Υ_1, outputs a message $m^\star$ and a time period $t^\star \in [\![1, \min(T_0, T_1)]\!]$. A challenge signature is then produced by flipping a coin $b \in \{0,1\}$ and signing the message under the key sk_b. The adversary is given this challenge signature $(\sigma^\star, r^\star, t^\star)$, and outputs a bit $b^\star$ at the end of the `guess` stage.

We denote by $q_{\mathcal{H}}$, $q_\hbar$, q_Σ, and q_Υ the number of queries from the random oracles, from the signing oracles, and from the receipt generating oracles and we assume that $T \geq \min(T_0, T_1)$ and $q_\Sigma \leq \ell$. The only requirement is that the challenge signature $(\sigma^\star, r^\star, t^\star)$ cannot be queried to a receipt generating oracle.

In any game Game_i, we denote by $\mathsf{Guess_i}$ the event $b^\star = b$. By definition,

$$|2\Pr[\mathsf{Guess_0}] - 1| = \mathbf{Adv}^{\mathsf{ano-cma}}_{\mathsf{TSCUSBM},\mathcal{A}}(k).$$

Game_1 First, we pick $(\alpha, \beta) \in [\![1, q-1]\!]^2$ at random and modify the simulation by replacing pk_0 by (X_1, Y_1) and pk_1 by $(\alpha X_1, \beta Y_1)$. The distributions of pk_0 and pk_1 are unchanged since we consider a random instance of the $(q_{\mathcal{H}}, T)$-xyz-DCAA problem. Therefore we have

$$\Pr[\mathsf{Guess_1}] = \Pr[\mathsf{Guess_0}].$$

Game_2 In this game, we simulate the random oracles $\mathcal{H}$ and $\hbar$ and maintain appropriate lists, which we denote by H-List and h-List. For any fresh query $(m, r) \in \{0,1\}^* \times \{0,1\}^{n_r}$

- to the oracle $\mathcal{H}$, we pick at random $s \in [\![1, q-1]\!]$, compute $sP \in \mathbb{G}$, store (m, r, s, sP, T) in the H-List and return sP as the answer to the oracle call;
- to the oracle $\hbar$, we pick at random $u \in [\![1, q-1]\!]$, store (m, r, u) in the h-List and returns u as the answer to the oracle call.

In the random oracle model, this game is clearly identical to the previous one. Hence, we obtain

$$\Pr[\mathsf{Guess_2}] = \Pr[\mathsf{Guess_1}].$$

Game_3 Now we simulate the signing oracles. We initialise a counter to $i = 1$, and for each new request $m \in \{0,1\}^*$, $t \in [\![1, T]\!]$ we pick $r \in \{0,1\}^{n_r}$ at random. If there exists a triple $(m, r, ?)$ in the h-List or a 5-tuple $(m, r, ?, ?, ?)$ in the H-List, we abort the simulation. We pick at random $s \in [\![1, q-1]\!]$, compute $sX_{T-t} \in \mathbb{G}$ and store (m, r, s, sX_{T-t}, t) in the H-List.

If the query is to Σ_0, the signature is (sR_i, r, t), we refresh the h-List with (m, r, h_i), then we increment the counter. If the query is to Σ_1 the signature is $(s\alpha^{t-1}\beta R_i, r, t)$, we refresh the h-List with $(m, r, \alpha h_i)$, then we increment the counter. This game perfectly simulates the signing oracle if we do not abort. As we abort with probability at most $(q_{\mathcal{H}} + q_\hbar)2^{-n_r}$, we have

$$|\Pr[\mathsf{Guess_3}] - \Pr[\mathsf{Guess_2}]| \leq \frac{q_\Sigma(q_{\mathcal{H}} + q_\hbar)}{2^{n_r}}$$

Game_4 We simulate the receipt generating oracles. When the adversary requests a couple message/putative signature $(m, (\sigma, r, t))$ on m to Υ_0 with $t \in [\![1, T]\!]$, we look in the H-List for a 5-tuple $(m, r, s, sX_{T-t'}, t')$, then the answer is the t-tuple $(sX_{T-t'+1}, \ldots, sX_{T-t'+t})$. If this request is on Υ_1, we answer with $(s\alpha^{T-t'+1}X_{T-t'+1}, \ldots, s\alpha^{T-t'+t}X_{T-t'+t})$ This simulation is perfect, therefore we have

$$\Pr[\mathsf{Guess_4}] = \Pr[\mathsf{Guess_3}].$$

Game_5 Finally, in this game, in the challenge generation: once the adversary $\mathcal{A}$ asks for a signature on a message $m^\star$ for a time period $t^\star$, we query the problem challenger with $t^\star$. It outputs $(Y_j)_{j \in [\![1, t^\star - 1]\!]} \in \mathbb{G}^{t^\star - 1}$ and $Q \in \mathbb{G}$. We pick a bit $b \in \{0,1\}$ and $r^\star \in \{0,1\}^{n_r}$ at random. If there exists a 5-tuple $(m^\star, r^\star, ?, ?, ?)$ in the H-List or a triple $(m^\star, r^\star, ?)$ in the h-List, we abort the simulation, else we update the H-List by storing $(m^\star, r^\star, \perp, Z, \perp)$ and the h-List by storing $(m, r, \alpha^b h)$. We output $((\alpha^{t^\star - 1}\beta)^b Q, r^\star, t^\star)$ as the challenge signature and $(Y_j)_{j \in [\![1, t^\star - 1]\!]}$ as the universal receipts for the challenge..

If $Q = Q_{\mathsf{real}} = x^T yz(x+h)^{-1}P$, this game perfectly simulates the challenge generation if we do not abort (which happens with probability at most $(q_{\mathcal{H}} + q_{\hbar})2^{-n_r}$). Therefore

$$|\Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{real}}] - \Pr[\mathsf{Guess}_4]| \leq \frac{q_{\mathcal{H}} + q_{\hbar}}{2^{n_r}}$$

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

$$\Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{random}}] \leq \frac{1}{2} + \frac{q_{\mathcal{H}} + q_{\hbar}}{2^{n_r}} + \frac{1}{2^k}.$$

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 $(q_{\mathcal{H}}, T) - xyz$-DCAA problem is:

$$\mathbf{Adv}^{(\mathsf{q}_{\mathcal{H}},\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}}_{\mathsf{Gen},\mathsf{Game}_5}(k) = |\Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{real}}] - \Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{random}}]|\,.$$

A simple computation gives the claimed bounds for ε' and τ'. □

Remark 7. *Using Boneh-Boyen's technique [2], it is possible to avoid the ideal hypothesis of perfect randomness of the hash function h.*

Corollary 2 (Anonymity of CUSBM and TSCUSBM+)**.** *Under the $(\ell, 1)$-xyz-DCAA assumption, the schemes CUSBM and TSCUSBM+ are Ano-CMA secure in the random oracle model.*

6 Final remarks and conclusion

Time-selective convertible signatures are introduced to eliminate the burden of registration of new public keys after the universal receipt publication. We formalised the security notions of a time-selective convertible undeniable signature scheme and thanks to a new technique to design cryptographic protocols achieving a tradeoff between authenticity and privacy, we proposed the first scheme meeting this definition.

The new schemes offer 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 convertible undeniable signature schemes.

The xyz-trick might have other applications: the use of this trick to design distributed contract signing and verifiable signature sharing protocols seems to be an interesting topic for further research.

Acknowledgements We express our gratitude to Pascal Paillier for his helpful comments.

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] D. Boneh and X. Boyen, *Short Signatures Without Random Oracles.*, Advances in Cryptology - EUROCRYPT 2004 (C. Cachin and J. Camenisch, eds.), Lect. Notes Comput. Sci., vol. 3027, Springer, 2004, pp. 56–73.

[3] ———, *Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups.*, J. Cryptology **21** (2008), no. 2, 149–177.

[4] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing.*, SIAM J. Comput. **32** (2003), no. 3, 586–615.

[5] D. Boneh, B. Lynn, and H. Shacham, *Short Signatures from the Weil Pairing.*, J. Cryptology **17** (2004), no. 4, 297–319.

[6] 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.

[7] C. Boyd and E. Foo, *Off-line Fair Payment Protocols using Convertible Signatures.*, Advances in Cryptology - ASIACRYPT'98 (K. Ohta and D. Pei, eds.), Lect. Notes Comput. Sci., vol. 1514, Springer, 1998, pp. 271–285.

[8] E. Bresson and J. Stern, *Proofs of Knowledge for Non-Monotone Discrete-Log Formulae and Applications.*, Information Security, ISC 2002 (A. H. Chan and V. D. Gligor, eds.), Lect. Notes Comput. Sci., vol. 2433, Springer, 2002, pp. 272–288.

[9] 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.

[10] J. Camenisch and M. Stadler, *Efficient Group Signature Schemes for Large Groups.*, Advances in Cryptology - CRYPTO'97 (B. S. Kaliski Jr., ed.), Lect. Notes Comput. Sci., vol. 1294, Springer, 1997, pp. 410–424.

[11] D. Chaum, *Zero-Knowledge Undeniable Signatures.*, Advances in Cryptology - EUROCRYPT'90 (I. Damgård, ed.), Lect. Notes Comput. Sci., vol. 473, Springer, 1991, pp. 458–464.

[12] D. Chaum and T. P. Pedersen, *Wallet Databases with Observers.*, Advances in Cryptology - CRYPTO'92 (E. F. Brickell, ed.), Lect. Notes Comput. Sci., vol. 740, Springer, 1993, pp. 89–105.

[13] 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.

[14] I. B. Damgård and T. P. Pedersen, *New Convertible Undeniable Signature Schemes.*, in Maurer [25], pp. 372–386.

[15] L. El Aimani and D. Vergnaud, *Gradually Convertible Undeniable Signatures.*, Applied Cryptography and Network Security, ACNS 2007 (J. Katz and M. Yung, eds.), Lect. Notes Comput. Sci., vol. 4521, Springer, 2007, pp. 478–496.

[16] 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.

[17] S. D. Galbraith, W. Mao, and K. G. Paterson, *RSA-based Undeniable Signatures for General Moduli.*, Topics in Cryptology - CT-RSA 2002 (B. Preneel, ed.), Lect. Notes Comput. Sci., vol. 2271, Springer, 2002, pp. 200–217.

[18] R. Gennaro, T. Rabin, and H. Krawczyk, *RSA-Based Undeniable Signatures.*, J. Cryptology **13** (2000), no. 4, 397–416.

[19] 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.

[20] X. Huang, Y. Mu, W. Susilo, and W. Wu, *Provably secure pairing-based convertible undeniable signature with short signature length*, Pairing 2007 (T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto, eds.), Lect. Notes Comput. Sci., vol. 4575, Springer, 2007, pp. 367–391.

[21] M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated Verifier Proofs and Their Applications.*, in Maurer [25], pp. 143–154.

[22] 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.

[23] F. Laguillaumie, P. Paillier, and D. Vergnaud, *Universally Convertible Directed Signatures*, Advances in Cryptology - ASIACRYPT 2005 (B. Roy, ed.), Lect. Notes Comput. Sci., vol. 3788, Springer, 2005, pp. 682–701.

[24] 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.

[25] 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.

[26] 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.

[27] M. Michels and M. Stadler, *Efficient Convertible Undeniable Signature Schemes.*, Selected Areas in Cryptography, 4th Annual International Workshop, SAC'97, 1997, pp. 231–244.

[28] S. Mitsunari, R. Sakai, and M. Kasahara, *A New Traitor Tracing.*, IEICE Trans. Fundamentals **E85-A** (2002), no. 2, 481–484.

[29] T. Ristenpart and S. Yilek, *The Power of Proofs-of-Possession: Securing Multiparty Signatures against Rogue-Key Attacks*, Advances in Cryptology - EUROCRYPT 2007 (M. Naor, ed.), Lect. Notes Comput. Sci., vol. 4515, Springer, 2007, pp. 228–245.

[30] 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.

[31] ———, *OAEP Reconsidered.*, J. Cryptology **15** (2002), no. 4, 223–249.

[32] D. Vergnaud, *Approximation diophantienne et courbes elliptiques. Protocoles asymétriques d'authentification non-transférable.*, Ph.D. thesis, Université de Caen, november 2006.

[33] F. Zhang, R. Safavi-Naini, and W. Susilo, *Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings.*, Progress in Cryptology - INDOCRYPT 2003 (T. Johansson and S. Maitra, eds.), Lect. Notes Comput. Sci., vol. 2904, Springer, 2003, pp. 191–204.

A Time-Selective Conversion for Huang *et al.*'s scheme

Let's first give a rough description of Huang, Mu, Susilo and Wei's scheme from [20].

Setup and Key Generation

Setup: Let k be a security parameter, Gen be a prime-order-BDH-parameter-generator and $(q, P, \mathbb{G}, \mathbb{H}, e)$ some output of $\mathsf{Gen}(k)$. Let $f_r : \mathbb{N} \to \mathbb{N}$ be a function. We denote $n_r = f_r(k)$. Let $[\{0,1\}^* \times \{0,1\}^{n_r} \longrightarrow \mathbb{G}]$ be a hash function family, and H_0 and H_1 be two random members of this family. The public parameters are $[(q, P, \mathbb{G}, \mathbb{H}, e), H_0, H_1]$

SKeyGen: Alice picks randomly two integers $a_1, a_2 \in [\![1, q-1]\!]$ and computes the points $P_1 = a_1 P$ and $P_2 = a_2 P$. Alice's public key is the pair (P_1, P_2) and her secret key is (a_1, a_2).

VKeyGen: the verifier Bob picks randomly an integer $y \in [\![1, q-1]\!]$ and computes the point $Y = yP$. Bob's public key is Y and his secret key is y.

Signing and controlling algorithm

Sign: Given a message $m \in \{0,1\}^*$, Alice picks at random $r \in \{0,1\}^{n_r}$ and computes the point

$$\sigma = (a_1 a_2) H_0(m||r) + a_1 H_1(m||r).$$

The convertible undeniable signature of the message m is (σ, r).

Control: Given a message m and a putative signature (σ, r), Alice checks that

$$e(\sigma, P)e(H_1(m, r), -P_1) = e(H_0(m||r), P_1)^{a_2}.$$

Confirmation / Denial protocols

Confirm: Given a message m and a signature (σ, r), Alice can confirm (σ, r) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P)e(H_1(m, r), -P_1) = e(H_0(m||r), P_1)^{a_2} \wedge P_2 = a_2 P).$$

Deny: Given a message m and an invalid signature (σ, r), Alice can deny (σ, r) with the following interactive proof of knowledge:

$$PK(a_2 : e(\sigma, P)e(H_1(m, r), -P_1) \neq e(H_0(m||r), P_1)^{a_2} \wedge P_2 = a_2 P).$$

Receipt generation and verification

Convert

- On input ϖ, Alice publishes the point $I = a_1 a_2 P$.

- Given a message $m \in \{0,1\}^*$ and a putative signature (σ, r) on m, Alice checks the validity of the signature thanks to the Control algorithm and then computes the point $\tilde{\sigma} = a_2 H_0(m||r) \in \mathbb{G}$. The individual receipt with respect to σ is $\tilde{\sigma}$.

Verify:

- The validity of the universal receipt I is decided by verifying that

  $$e(P_1, P_2) = e(I, P).$$

  If it is valid, given a signature (σ, r) on a message $m \in \{0,1\}^*$ and I, everyone checks the validity of this signature by verifying that

  $$e(\sigma, P)e(H_1(m, r), -P_1) = e(H_0(m||r), I).$$

- Given a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m and a putative individual receipt $\tilde{\sigma}$ on (σ, r), the validity of the receipt is decided by checking whether

  $$e(\tilde{\sigma}, P) = e(P_2, H_0(m||r))$$

  or not. If $\tilde{\sigma}$ is valid, then the validity of (σ, r) is decided by checking whether

  $$e(\sigma, P)e(H_1(m, r), -P_1) = e(\tilde{\sigma}, P_2)$$

  or not.

To obtain a logarithmic size of the time-selective receipts, the following procedures must be included:

Sign: Given a message $m \in \{0,1\}^*$, and a time period $t \in [\![1, T]\!]$. Alice picks at random $r \in \{0,1\}^{n_r}$ and computes the point

$$\sigma = (a_1 a_2^t) H_0(m||r) + a_1 H_1(m||r).$$

The convertible undeniable signature of the message m is (σ, r).

Convert

- On input ϖ, and an integer $t \in [\![1, T]\!]$, this protocol consists for Alice running $\mathsf{ReceiptGen}(P_1, a_2, t)$.

- Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$, a putative signature (σ, r) on m for the time period t, Alice checks the validity of the signature thanks to the Control algorithm and then runs $\mathsf{ReceiptGen}(H(m||r), a_2, t)$.