Time-Selective Convertible Undeniable Signatures

Fabien Laguillaumie[1,2] and Damien Vergnaud[2]

[1] France Telecom Research and Development
42, rue des Coutures, B.P. 6243, 14066 Caen Cedex 4, France,
[2] Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, Campus II, B.P. 5186,
14032 Caen Cedex, France,
{laguillaumie,vergnaud}@math.unicaen.fr

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. 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 notion as the *time-selective conversion.*

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

1 Introduction

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

For privacy reasons, it is preferable, 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 at the very end of the eighties by Chaum and van Antwerpen [14]. In this setting, the verification (and the denial) of a signature requires the cooperation of the signer. The security of their protocol relies on the discrete logarithm problem, but suffers from the fact that the interactive protocols were not zero-knowledge. One year later, Chaum improved significantly the initial proposal in [12] by providing a zero-knowledge version. Moreover 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*, proposed in [6] 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 [25] who proposed a repaired version with heuristic security. Since then, many schemes have then been proposed, based upon classical signatures, such as Schnorr [26], El Gamal [17] and RSA [18–20]. Very recently, Monnerat and Vaudenay [28] proposed short undeniable signatures based on the computation of characters which do not provide the conversion property. Convertible undeniable signatures have given rise to many applications in cryptography [6, 7, 16].

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. Besides, in the identity-based paradigm[1], the problem is even more serious since there is a unique secret key by identity.

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 [6] 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 [18] 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 the two worlds and, introducing Zhang, Safavi-Naini and Susilo's technique from [33,34], we obtain a convertible undeniable signature protocol which is one of the most efficient. As in [26], the signer not only can selectively convert valid signatures into ordinary digital signatures, but he can also convert any invalid signature into an universally verifiable statement about this fact. Moreover, to overcome the difficulty mentionned above, we introduce and formalize the *time-selective convertible undeniable signatures* which supports

[1] An identity-based undeniable signature scheme has been proposed by Libert and Quisquater in 2004 [24], and, like our scheme, it is built with bilinear maps. However it does not provide the universal conversion property.

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 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 decisional assumption.

The article is organized as follows: first we formally define the concept of time-selective convertible undeniable signature scheme and its security model. Then, we review the cryptographic properties of bilinear maps and describe the problems upon which depend our schemes. In the following section, we describe our new scheme and its time-selective convertible variant, and finally we prove its security in the random oracle model.

2 Formal definition and security model

2.1 Definition

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

Definition 1 (Time-Selective Convertible Undeniable Signature). *Given an integer k, a* time-selective convertible undeniable signature scheme *TSCUS with security parameter k is defined by the following:*

- *a* **common parameter generation algorithm** *TSCUS.*`Setup`*: it is a probabilistic algorithm which takes as input k. The outputs are* the public parameters*;*
- *a* **key generation algorithm** *TSCUS.*`KeyGen`*: it is a probabilistic algorithm which takes as inputs the public parameters and outputs a pair of keys (pk, sk) and a public number of time periods $T \in \mathbb{N}$;*
- *a* **signing algorithm** *TSCUS.*`Sign`*: it is a probabilistic algorithm which takes as inputs T, a message m, a secret key sk, an integer $t \in [\![1, T]\!]$, and the public parameters. The output σ is* a time-selective convertible undeniable signature on *m for the time period t;*
- **confirming/denying protocols** *TSCUS.$\{$*`Confirm`*,* `Deny`*$\}$: they are protocols which take as inputs T, a message m, an integer $t \in [\![1, T]\!]$, a bit string σ, a pair of keys (pk, sk) and the public parameters. The output is a (possibly non-interactive) non-transferable proof that σ is actually a valid/invalid time-selective convertible undeniable signature on m for the time period t, with respect to the key pk;*
- *an* **individual receipt generation algorithm** *TSCUS.*`IReceipt`*: it is an algorithm which takes as inputs T, a message m, an integer $t \in [\![1, T]\!]$, a bit string σ, a secret key sk and the public parameters. It outputs an* individual receipt *$\tilde{\sigma}$ which makes it possible to universally verify whether σ is valid or not.*

- *a* **verifying algorithm for individually converted signature** *TSCUS.IVerify: it is a deterministic algorithm which takes as inputs T, a message m, an integer $t \in [\![1, T]\!]$, a bit string σ, a bit string $\tilde{\sigma}$, the signer's public key pk, and the public parameters. It tests whether $\tilde{\sigma}$ is a valid individual receipt with respect to σ and the public key pk. If it does, the algorithm states whether σ is a valid time-selective convertible undeniable signature on m for the time period t with respect to the key pk or not, else it outputs Error;*
- *a* **universal receipt generation algorithm** *TSCUS.UReceipt: it is a deterministic algorithm which takes as inputs T, a secret key sk, an integer $t \in [\![1, T]\!]$, and the public parameters and outputs a universal receipt $\mathcal{I}_t$ which makes it possible to universally verify all time-selective convertible undeniable signature σ on m for any time period $t' \leq t$;*
- *a* **verifying algorithm for universally converted signature** *TSCUS.UVerify: it is a deterministic algorithm which takes as inputs T, a message m, two integers $(t, t') \in [\![1, T]\!]^2$ such that $t' \leq t$, a bit string σ, a public key pk, a bit string $\mathcal{I}_t$ and the public parameters. It tests whether $\mathcal{I}_t$ is a valid universal receipt for the time period t with respect to the key pk. If it does, it states whether σ is a valid time-selective convertible undeniable signature on m for the time period t' with respect to the key pk or not, else it outputs Error;*

and must satisfy the following properties (formally discussed in the next section):

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. unforgeability: *given a public key pk, it is computationally infeasible, without the knowledge of the corresponding secret key to produce a time-selective convertible undeniable signature for any time period $t \in [\![1, T]\!]$ which is accepted by the verification algorithms or by the confirming protocols;*
3. anonymity: *given a message m and a time-selective convertible undeniable signature σ on m for a time period $t' \in [\![1, T]\!]$, it is computationally infeasible without the knowledge of the signing secret key or of some universal receipt $\mathcal{I}_t$ for some $t \geq t'$, to determine which secret key sk was used to generate σ;*
4. non-transferability: *a verifier participating in an execution of the confirming/denying protocols does not obtain information that could be used to convince a third party about the validity/invalidity of a signature.*

Definition 2 (Convertible Undeniable Signature). *Given an integer k, a* convertible undeniable signature scheme *with security parameter k is a time-selective convertible undeniable signature scheme with security parameter k, and whose key generation algorithm always outputs 1 as the number of time periods.*

2.2 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 [2]. In this model, hash functions are idealized as oracles which output a random value for each new query.

Security against existential forgery under chosen message attack The *de facto* standard notion of security for digital signatures was defined by Goldwasser, Micali and Rivest [21] as *existential forgery against adaptive chosen message attacks* (EF-CMA). For time-selective convertible undeniable signatures, the unforgeability security is defined along the same lines, with the notable difference that, while mounting a chosen-message attack, we suppose that the adversary has access to the universal receipts $\mathcal{I}_t$ for all $t \in [\![1, T]\!]$. Moreover, he is allowed to query a receipt generating oracle Υ and a confirming/denying oracle Ξ on any couple message/signature of his choice, in addition to the classical access to the signing oracle Σ and to the random oracle $\mathcal{H}$. As usual, in the adversary answer, there is the natural restriction that in the returned triple message/signature/time period $(m^\star, \sigma^\star, t^\star)$, the signature $\sigma^\star$ on $m^\star$ has not been obtained from Σ for the time period $t^\star$. However, $\mathcal{A}$ is allowed to query the signing oracle on the returned message $m^\star$ in any time period $t \in [\![1, T]\!]$ (especially for $t = t^\star$).

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

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

$\text{params} \xleftarrow{R} \textit{TSCUS.Setup}(k)$
$(pk, sk, T) \xleftarrow{R} \textit{TSCUS.KeyGen}(\text{params})$
For j from 1 to T do $\mathcal{I}_j \leftarrow \textit{TSCUS.UConvert}(\text{params}, pk, sk, j)$
$(m^\star, \sigma^\star, t^\star) \leftarrow \mathcal{A}^{\mathcal{H},\Sigma,\Upsilon,\Xi}(\text{params}, pk, \{\mathcal{I}_j\}_{j \in [\![1,T]\!]})$
Return $\texttt{TSCUS}.\textit{UVerify}(\text{params}, pk, m^\star, \sigma^\star, t^\star, T, \mathcal{I}_T)$

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

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

Anonymity In this section, 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. Our formalization follows the notion, introduced by Galbraith and Mao in [18] for undeniable signatures. We consider an Ano-CMA-adversary $\mathcal{A}$ that runs in two stages. In the `find` stage, it takes as input two public keys pk_0 and pk_1 and outputs a message $m^\star$, a time period $t^\star$ together with some state information s. In the `guess` stage it gets a challenge time-selective convertible undeniable signature $\sigma^\star$ formed by signing the message $m^\star$ at random under one of the two keys for the time period $t^\star$, and must say which key was chosen. In both stages, the adversary has access to the signing oracles Σ_0, Σ_1, to the receipt generating oracles Υ_0 and Υ_1 to the confirming/denying oracles Ξ_0 and Ξ_1, and to the random oracle(s) $\mathcal{H}$. The attacker is also given the universal receipts $\mathcal{I}_1, \ldots, \mathcal{I}_{t^\star-1}$ of both potential signers. The only restriction of the attacker is that he cannot query the couple $(m^\star, \sigma^\star)$ on the converting and confirming/denying oracles.

Definition 4 (Anonymity). *Let* TSCUS *be a time-selective convertible undeniable signature scheme and let* $\mathcal{A}$ *be an* Ano-CMA*-adversary against* TSCUS*. We consider the following random experiments, for* $r \in \{0,1\}$*, where* k *is a security parameter:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{\textsf{ano-cma}-r}_{\textsf{TSCUS},\mathcal{A}}(k)}$

$\text{params} \xleftarrow{R} \texttt{TSCUS.Setup}(k)$
$(pk_0, sk_0, T_0) \xleftarrow{R} \texttt{TSCUS.KeyGen}(\text{params})$
$(pk_1, sk_1, T_1) \xleftarrow{R} \texttt{TSCUS.KeyGen}(\text{params})$
$(m^\star, t^\star, s) \leftarrow \mathcal{A}^{\mathcal{H},\Sigma_0,\Sigma_1,\Upsilon_0,\Upsilon_1,\Xi_0,\Xi_1}(\texttt{find}, \text{params}, pk_0, pk_1)$ *with* $t^\star \leq \min(T_0, T_1)$
$\sigma^\star \leftarrow \texttt{TSCUS.Sign}(\text{params}, m, sk_r, t^\star)$
For j *from* 1 *to* $t^\star - 1$ *do*
  $\mathcal{I}_j^0 \leftarrow \texttt{TSCUS.UConvert}(\text{params}, pk_0, sk_0, j)$
  $\mathcal{I}_j^1 \leftarrow \texttt{TSCUS.UConvert}(\text{params}, pk_1, sk_1, j)$
$d \leftarrow \mathcal{A}^{\mathcal{H},\Sigma_0,\Sigma_1,\Upsilon_0,\Upsilon_1,\Xi_0,\Xi_1}(\texttt{guess}, \text{params}, pk_0, pk_1, m^\star, \sigma^\star, t^\star, s, \{\mathcal{I}_j^0, \mathcal{I}_j^1\}_{j \in [\![1, t^\star-1]\!]})$
Return d

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

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

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

Remark 1. 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, which simulation is impossible in the random oracle model [18]. Usually [18,29], this problem is overcome with the use of designated-verifier proofs [22]. 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). We denote, as in [11], the zero-knowledge proof of equality of the discrete logarithm of $Y \in \mathbb{G}$ in base P and the one of $y \in \mathbb{H}$ in base g by $PK(a : y = g^a \wedge Y = aP)$. We use the notation $PK(a : y \neq g^a \wedge Y = aP)$ for the proof of inequality of the discrete logarithms. We refer the reader to Chaum and Pedersen's paper [16] for the proof of equality, and these of Camenisch and Shoup [10] for the proof of inequality.

In the design of the new scheme, we also need zero-knowledge proofs of equality/inequality of a root of a discrete logarithm: given $t \in \mathbb{N}$, $PK(a : y = g^{a^t} \wedge Y = aP)$ and $PK(a : y \neq g^{a^t} \wedge Y = aP)$. Efficient protocols can be found in [8, 11].

3.2 Bilinear maps

Recently 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].

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 (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 difficulty 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 mentionned 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 [3] 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[2] introduced by Mitsunari, Sakai, Kasahara in 2002 in relation with the security of a traitor tracing scheme [27]. Using the tricky polynomial construction from [27], it is easy to prove that the ℓ-CAA problem is polynomial time equivalent[3] to the:

(ℓ, T)–**Computational CAA Problem ((ℓ, T)-CCAA):** Let $(\mathbb{G}, +)$ be a group of prime order q, and let x be in $[\![1, q-1]\!]$. Given two integers ℓ and T and

$$\left[(x^i P)_{i \in [\![0, 2T-1]\!]}, \left(\frac{x^T}{x + h_j} P, h_j \right)_{j \in [\![1, \ell]\!]} \right] \in \mathbb{G}^{2T} \times \left(\mathbb{G} \times [\![1, q-1]\!] \right)^{\ell},$$

compute a pair $(Q, h) \in \mathbb{G} \times ([\![1, q-1]\!] \setminus \{h_1, \dots, h_\ell\})$ verifying $(x+h)Q = x^T P$.

[2] collusion attack algorithm with ℓ traitors

[3] In fact, the ℓ-CAA problem from [27] is exactly the $(\ell, 1)$-CCAA problem

Like the DDH problem, thanks to the bilinear map, the decisional problem associated to (ℓ, T)-CCAA is easy. Therefore, we introduce a decisional variant of (ℓ, T)-CCAA, 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 $h \notin \{h_1, \dots, h_\ell\}$

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 [27] is analogous to the one between the xyz-DDH problem and the CDH problem. We want to stress that, though non-standard, these problems are random-self reducible and achieve generic security (as defined by Shoup [31]). The proof is similar to the one of the generic security of ℓ-SDH [3].

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

Definition 7 ((ℓ, T)-xyz-DCAA assumption). *Let $\mathcal{G}en$ 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 $\mathcal{G}en$ 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 $r \in \{0, 1\}$:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{(\ell,\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}-\mathsf{r}}_{\mathcal{G}en,D}(k)}$

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

We define the corresponding advantage *of D in solving the $(\ell, T) - xyz$-DCAA problem* via*:* $\mathbf{Adv}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}}_{\mathcal{G}en,D}(k) = \left|\Pr\left[\mathbf{Exp}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}\text{-}0}_{\mathcal{G}en,D}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{(\ell,\mathsf{T})\text{-}\mathsf{xyz}\text{-}\mathsf{dcaa}\text{-}1}_{\mathcal{G}en,D}(k) = 1\right]\right|$.

Given $(k,\tau) \in \mathbb{N}^2$ *and* $\varepsilon \in [0,1]$, $\mathcal{G}en$ *is said to be* (k,τ,ε)*-*(ℓ,T)*-xyz-DCAA-secure if no adversary* D *running in time* τ *has advantage* $\mathbf{Adv}^{(\ell,\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}}_{\mathcal{G}en,D}(k) \geq \varepsilon$.

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, $\mathcal{G}en$ be a prime-order-BDH-parameter-generator and $(q,P,\mathbb{G},\mathbb{H},e)$ some output of $\mathcal{G}en(k)$. Let $f_r : \mathbb{N} \rightarrow \mathbb{N}$ 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 $[(q,P,\mathbb{G},\mathbb{H},e),H,h]$

KeyGen: Alice picks randomly two integers $a_1, a_2 \in [\![1,q-1]\!]$ and computes the points $P_1 = a_1P$ and $P_2 = a_2P$. Alice's public key is the pair (P_1,P_2) and her secret key is (a_1,a_2).

Signing algorithm

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

$$\sigma = \frac{a_1a_2}{(a_2 + h(m||r))}H(m||r).$$

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

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: $PK(a_2 : e(\sigma, P_2+h(m||r)P) = e(H(m||r),P_1)^{a_2} \wedge P_2 = a_2P)$.

Deny: Given a message m and an invalid signature (σ,r), Alice can deny (σ,r) with the following interactive proof of knowledge: $PK(a_2 : e(\sigma, P_2+h(m||r)P) \neq e(H(m||r),P_1)^{a_2} \wedge P_2 = a_2P)$.

Receipt generation and verification

IReceipt: Given a message $m \in \{0,1\}^*$ and a putative signature (σ,r) on m, Alice computes the point $\tilde{\sigma} = a_2H(m||r) \in \mathbb{G}$. The individual receipt with respect to σ is $\tilde{\sigma}$.

IVerify: 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.

UReceipt: Alice publishes the point $I = a_1 a_2 P$.

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

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 practice, the size of an element of $\mathbb{G}$ can be reduced by a factor 2 with compression techniques and the random salt has size $n_r = 112$. Therefore, the size of the signature is only 272 bits. Furthermore, a receipt (individual and universal) is also an element of $\mathbb{G}$, and therefore has bit size 160. 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 The first 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`

KeyGen: 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).

Signing 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).

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

IReceipt: Given an integer $t \in [\![1, T]\!]$, a message $m \in \{0,1\}^*$ and a signature (σ, r) on m for the time period t, Alice computes the t-tuple $\tilde{\sigma} = (a_2 H(m||r), a_2^2 H(m||r), \dots, a_2^t H(m||r))$. The individual receipt with respect to σ is $\tilde{\sigma}$.

IVerify: 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 $\tilde{\sigma} = (\tilde{\sigma}^{(1)}, \dots, \tilde{\sigma}^{(t)})$ 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.

UReceipt: Given an integer $t \in [\![1, T]\!]$, this protocol consists for Alice in publishing the t-tuple $I_t = (a_1 a_2 P, a_1 a_2^2 P, \dots, a_1 a_2^t P) \in \mathbb{G}^t$.

UVerify: Given an integer $t \in [\![1, T]\!]$, the validity of the universal receipt $I_t = (I^{(1)}, \dots, I^{(t)})$ is decided by verifying that $e(P_1, P_2) = e(I^{(1)}, P)$ and $e(I^{(i-1)}, P_2) = e(I^{(i)}, P)$ 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 $e(\sigma, P_2 + h(m||r)P) = e(H(m||r), I^{(t')})$.

Remark 2. 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.

5 Security results

Unforgeability The theorem below states that TSCUSBM is EF-CMA secure assuming the intractibility 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 $\mathcal{G}en$ 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 CUSBM in the random oracle model, that produces an existential forgery with probability $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{CUSBM},\mathcal{A}}$, within time τ, making $q_{\mathcal{H}}$, q_{Σ} and 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]. For the reader's convenience, it is supplied in appendix A.

Corollary 1 (Unforgeability of CUSBM). *Under the CDH assumption, the scheme CUSBM is EF-CMA secure in the random oracle model.*

Remark 3. 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 [27] assumption.

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 $\mathcal{G}en$ 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 $\mathbf{Adv}^{\mathsf{ano-cma}}_{CUS,\mathcal{A}}$, 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 [32]: we define a sequence of game Game_0, ..., 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 $\mathcal{G}en(k)$ and $\left[(X_i)_{i\in[\![1,2T-1]\!]}, Y, Z, h), (R_j, h_j)_{j\in[\![1,\ell]\!]}\right]$ in $\mathbb{G}^{2T+1} \times [\![1, q-1]\!] \times (\mathbb{G} \times [\![1, q-1]\!])^{\ell}$ 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,

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

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 $\mathsf{H\text{-}List}$ and $\mathsf{h\text{-}List}$. 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 $\mathsf{H\text{-}List}$ 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 $\mathsf{h\text{-}List}$ 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 initialize 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 $\mathsf{h\text{-}List}$ or a 5-tuple $(m, r, ?, ?, ?)$ in the $\mathsf{H\text{-}List}$, 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 $\mathsf{H\text{-}List}$.
If the query is to Σ_0, the signature is (sR_i, r, t), we refresh the $\mathsf{h\text{-}List}$ 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 $\mathsf{h\text{-}List}$ with $(m, r, \alpha h_i)$, then we incremente 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 $\mathsf{H\text{-}List}$ 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 $\mathsf{H\text{-}List}$ or a triple $(m^\star, r^\star, ?)$ in the $\mathsf{h\text{-}List}$, we abort the simulation, else we update the $\mathsf{H\text{-}List}$ by storing $(m^\star, r^\star, \perp, Z, \perp)$ and the $\mathsf{h\text{-}List}$ 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: $\mathbf{Adv}^{(q_{\mathcal{H}},\mathsf{T})-\mathsf{xyz}-\mathsf{dcaa}}_{\mathcal{G}en,\mathsf{Game}_5}(k) = |\Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{real}}] - \Pr[\mathsf{Guess}_5|Q = Q_{\mathsf{random}}]|$. A simple computation gives the claimed bounds for ε' and τ'. $\square$

Remark 4. Using Boneh-Boyen's technique [3], it is possible to avoid the ideal hypothesis of perfect randomness of the hash function h.

Corollary 2 (Anonymity of CUSBM). *Under the $(\ell, 1)$-xyz-DCAA assumption, the scheme CUSBM is Ano-CMA secure in the random oracle model.*

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 formalized 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-technique* has other applications. For example, we are also able to design very efficient universally convertible directed signatures – such a construction had remained open since 1993. Full details will appear elsewhere. Other applications will certainly appear: the use of this trick to design distributed contract signing and verifiable signature sharing protocols seems to be an interesting topic for further research.

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

References

1. P. S. L. M. Barreto, H. Y. Kim: Fast hashing onto elliptic curves over fields of characteristic 3, Cryptology ePrint Archive, Report 2001/098 (2001)
2. M. Bellare, P. Rogaway: Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. Proc. of 1st ACM Conference on Computer and Communications Security. 62–73 (1993)
3. D. Boneh, X. Boyen: Short Signatures Without Random Oracles. Proc. of Eurocrypt'04, Springer LNCS Vol. 3027, 56–73 (2004)
4. D. Boneh, M. Franklin: Identity-based Encryption from the Weil Pairing. SIAM J. Computing, 32(3), 586–615 (2003)
5. D. Boneh, B. Lynn, H. Shacham: Short signatures from the Weil pairing. Proc. of Asiacrypt'01, Springer LNCS Vol. 2248, 514–532 (2001)
6. J. Boyar, D. Chaum, I. B. Damgård, T.P. Pedersen: Convertible undeniable signatures. Proc. of Crypto'90, Springer LNCS Vol. 537, 189–205 (1991)
7. C. Boyd, E. Foo: Off-line Fair Payment Protocols using Convertible Signatures. Proc. of Asiacrypt'98, Springer LNCS Vol. 1514, 271–285 (1998)
8. E. Bresson and J. Stern: Proofs of Knowledge for Non-Monotone Discrete-Log Formulae and Applications. Proc. of ISC 2002, Springer LNCS Vol. 2433, 272–288 (2002)
9. J. Camenisch, M. Michels: Confirmer Signature Schemes Secure against Adaptative Adversaries. Proc. of Eurocrypt'00, Springer LNCS Vol. 1807, 243–258 (2000)
10. J. Camenisch, V. Shoup: Practical Verifiable Encryption and Decryption of Discrete Logarithms. Proc. of Crypto'03, Springer LNCS Vol. 2729, 126–144 (2003)
11. J. Camenisch, M. Stadler: Efficient Group Signature Schemes for Large Groups. Proc. of Crypto'97, Springer LNCS Vol. 1296, 410–424 (1997)
12. D. Chaum: Zero-Knowledge undeniable signatures. Proc. of Eurocrypt'90, Springer LNCS Vol. 473, 458–464 (1991)
13. D. Chaum: Designated Confirmer Signatures. Proc. of Eurocrypt'94, Springer LNCS Vol. 950, 86–91 (1995)
14. D. Chaum, H. van Antwerpen: Undeniable Signatures. Proc. of Crypto'89, Springer LNCS Vol. 435, 212–216 (1989)
15. D. Chaum, E. van Heijst, and B. Pfitzmann: Cryptographically strong undeniable signatures, unconditionally secure for the signer. Proc. of Crypto'91, Springer LNCS Vol. 576, 470–484 (1992)

16. D. Chaum, T.P. Pedersen: Wallet Databases with Observers. Proc. of Crypto'92, Springer LNCS Vol. 740, 89–105 (1993)
17. I. Damgard, T.P. Pedersen: New convertible undeniable signature schemes. Proc. of Eurocrypt'96, Springer LNCS Vol. 1070, 372—386 (1996)
18. S. Galbraith, W. Mao: Invisibility and anonymity of undeniable and confirmer signatures. Proc. of CT-RSA 2003, Springer LNCS Vol. 2612 80–97 (2003)
19. S. Galbraith, W. Mao, K.G. Paterson: RSA-based undeniable signatures for general moduli. Proc. of CT-RSA 2002, Springer LNCS Vol. 2271, 200–217 (2002)
20. R. Gennaro, H. Krawczyk, T. Rabin: RSA-based undeniable signatures. Proc. of Crypto'97, Springer LNCS Vol. 1294, 132–149 (1997)
21. S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. SIAM J. Computing, 17 (2), 281–308 (1988)
22. M. Jakobsson, K. Sako, R. Impagliazzo: Designated Verifier Proofs and their Applications. Proc.of Eurocrypt'96, Springer LNCS Vol. 1070, 142–154 (1996)
23. H. Krawczyk, T. Rabin: Chameleon Signatures. Proc. of NDSS 2000, 143–154 (2000)
24. B. Libert, J.-J. Quisquater: Identity Based Undeniable Signatures. Proc. of CT-RSA 2004, Springer LNCS Vol. 2964, 112–125 (2004)
25. M. Michels, H. Petersen, P. Horster: Breaking and repairing a convertible undeniable signature scheme. Proc. of ACM Conference on Computer and Communications Security 1996, 148–152, ACM Press (1996)
26. M. Michels, M. Stadler: Efficient Convertible Undeniable Signature Schemes. Proc. of SAC'97, 231–244 (1997)
27. S. Mitsunary, R. Sakai, M. Kasahara: A New Traitor Tracing. IEICE Trans. Fundamentals, Vol. E85-A (2), 481–484 (2002)
28. J. Monnerat, S. Vaudenay: Undeniable Signatures Based on Characters: How to Sign with One Bit. Proc. of PKC 2004, Springer LNCS Vol. 2947, 69–85 (2004)
29. D. Pointcheval: Self-Scrambling Anonymizers. Proc. of Financial Cryptography 2000, Springer LNCS Vol. 1962, 259–275 (2000)
30. D. Pointcheval, J. Stern: Security Arguments for Digital Signatures and Blind Signatures. J. Cryptology, Vol. 13 (3), 361–396 (2000)
31. V. Shoup: Lower bounds for discrete logarithms and related problems. Proc. of Eurocrypt'97, Springer LNCS Vol. 1233, 256–266 (1997)
32. V. Shoup: OAEP reconsidered. Manuscript, November 16, 2000. Revised September 18, 2001. Full length version of the extended abstract in Proc. Crypto'01 (2001)
33. F. Zhang, R. Safavi-Naini, W. Susilo: Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings. Proc. of Indocrypt 2003, Springer LNCS Vol. 2904, 191–204 (2003) Revised version available from the authors.
34. F. Zhang, R. Safavi-Naini, W. Susilo: An efficient Signature Scheme from Bilinear Pairings and its Applications. Proc. of PKC 2004, Springer LNCS Vol. 2947, 277–290 (2004)

A Proof of unforgeability

We define a sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_3$ 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 $\mathcal{G}en(k)$ 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 $\mathsf{Forge_j}$ the event $\mathsf{TSCUSBM}.\mathtt{UVerify}(m^\star, (\sigma^\star, r^\star, t^\star), pk, \mathcal{I}_T) = 1$. 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_2P) and the T-tuple $(\mathcal{I}_1, \ldots, \mathcal{I}_T)$ by $(a_2X, a_2^2X, \ldots, a_2^TX)$. 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 returns 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 outputs 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. □