Identity Based Undeniable Signatures

Benoît Libert Jean-Jacques Quisquater

UCL Crypto Group
Place du Levant, 3. B-1348 Louvain-La-Neuve. Belgium
{libert,jjq}@dice.ucl.ac.be –– http://www.uclcrypto.org/

Abstract. In this paper, we give a first example of identity based undeniable signature using pairings over elliptic curves. We extend to the identity based setting the security model for the notions of invisibility and anonymity given by Galbraith and Mao in 2003 and we prove that our scheme is existentially unforgeable under the Bilinear Diffie-Hellman assumption in the random oracle model. We also prove that it has the invisibility property under the Decisional Bilinear Diffie-Hellman assumption and we discuss about the efficiency of the scheme.

Keywords. ID-based cryptography, undeniable signatures, pairings, provable security.

1 Introduction

Identity based public key cryptography is a paradigm proposed by Shamir in 1984 ([38]) to simplify key management and remove the necessity of public key certificates. To achieve this, the trick is to let the user's public key be an information identifying him in a non ambiguous way (e-mail address, IP address, social security number...). The removal of certificates allows avoiding the trust problems encountered in today's certificate-based public key infrastructures (PKIs). This kind of cryptosystem involves trusted authorities called private key generators (PKGs) that have to deliver private keys to users after having computed them from their identity information (users do not generate their key pairs themselves) and from a master secret key. End-users do not have to enquire for a certificate for their public key. The only things that still have to be certified are the public keys of trusted authorities (PKGs) since they are involved in every encryption or signature verification processes. Although, this does not completely remove the need for certificates, this need is drastically reduced since many users depend on the same authority. Several practical identity based signature schemes (IBS) have appeared since 1984 ([24],[37]) but a practical identity based encryption scheme (IBE) was only found in 2001 ([4]) by Boneh and Franklin who took advantage of the properties of suitable bilinear maps (the Weil or Tate pairing) over supersingular elliptic curves. Many other identity based primitives based on pairings were proposed after 2001: digital signatures, authenticated key exchange, non-interactive key agreement, blind and ring signatures, signcryption,. . . ([7],[9],[16],[26],[33],[39],[40],. . .).

Undeniable signatures are a concept introduced by Chaum and van Antwerpen in 1989 ([11]). It is a kind of signatures that cannot be verified without interacting with the signer. They are useful in situations where the validity of a signature must not be universally verifiable. For example, a software vendor might want to embed signatures into his products and allow only paying customers to check the authenticity of these products. If the vendor actually signed a message, he must be able to convince the customer of this fact using a confirmation protocol and, if he did not, he must also be able to convince the customer that he is not the signer with a denial protocol. These proofs have to be non-transferable: once a verifier is convinced that the vendor did or did not sign a message, he should be unable to transmit this conviction to a third party.

In some applications, a signer needs to decide not only when but also by whom his signatures can be verified. For example a voting center can give a voter a proof that his vote was actually counted without letting him the opportunity to convince someone else of his vote. That is the motivation of designated verifier proofs for undeniable signatures. This kind of proof involves the verifier's public key in such a way that he is not able to convince a third party that a signer actually signed a message or not because he is able to produce such a valid proof himself using his private key. Several proof systems were proposed for undeniable signatures ([19],[27],[34],...). The use of designated verifier proofs ([27]) can provide non-interactive and non-transferable confirmation and denial protocols.

Several examples of undeniable signature schemes based on discrete logarithm were proposed ([11],[12],[13]) and the original construction of Chaum and van Antwerpen ([11]) was proven secure in 2001 by Okamoto and Pointcheval ([32]) thanks to full domain hash techniques and the use of a new kind of computational problems. Several convertible [1] undeniable signatures were proposed ([6],[17],[30],...). In 1997, Michels and Stadler proposed a convertible undeniable signature supporting designated-verifier verification. RSA-based undeniable signatures were designed by Gennaro, Krawczyk and Rabin ([22]) and Galbraith, Mao and Paterson ([20]). However, no secure identity based undeniable signature has been proposed so far. A solution was proposed in [25] but it was shown in [41] to be insecure. In this paper, we show how bilinear maps over elliptic curves can provide such a provably secure scheme. It is known ([31]) that an undeniable signature can be built from any public key encryption scheme and a similar result is likely to hold in the ID-based setting. However, the scheme described here can offer a security that is more tightly related to some computational problem than a scheme derived from the Boneh-Franklin IBE ([4]).

Chaum, van Heijst and Pfitzmann introduced the notion of 'invisibility' for undeniable signatures. Intuitively, it corresponds to the inability for a distinguisher to decide whether a message-signature pair is valid for a given user or not. The RSA-based schemes described in [20] and [22] do not provide invisibility. In [21], Galbraith and Mao describe a new RSA-based undeniable signature that provides invisibility under the so-called composite decision Diffie-Hellman assumption and they show that invisibility and anonymity [2] are essentially equivalent security notions for undeniable signature schemes satisfying some particular conditions. In this paper, we extend these two security notions to the identity based setting and we prove in the random oracle model that our scheme is both existentially unforgeable and invisible under some reasonable computational assumptions. Invisibility and anonymity can also be shown to be equivalent in the context of identity based cryptography and we will not do it here.

In section 2, we first recall the properties of pairings over elliptic curves before formally describing security notions related to identity based undeniable signatures. In section 3, we describe the different components of our scheme. We then show their correctness and we discuss about their efficiency. The rest of the paper is made of a security analysis of the scheme in the random oracle model.

2 Preliminaries

2.1 Overview of pairings and bilinear problems

Let us consider groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q. We need a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

1. Bilinearity: $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_q$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
2. Non-degeneracy: for any $P \in \mathbb{G}_1$, $\hat{e}(P, Q) = 1$ for all $Q \in \mathbb{G}_1$ iff $P = \mathcal{O}$.
3. Computability: some efficient algorithm can compute $\hat{e}(P, Q)\ \forall\ P, Q \in \mathbb{G}_1$.

[1] See [6]. Convertible undeniable signatures are undeniable signatures that can be converted by the signer into universally verifiable signatures.

[2] This security notion is related to the inability for an adversary to decide which user generated a particular message-signature pair in a multi-user setting.

$\mathbb{G}_1$ is a subgroup of the additive group of points of a supersingular elliptic curve $E(\mathbb{F}_p)$ over a finite field. $\mathbb{G}_2$ is a cyclic subgroup of the multiplicative group associated to a finite extension of $\mathbb{F}_p$. The original Weil pairing ([29]) is defined over non-cyclic groups. For our purposes, we use pairings over cyclic groups such as the modified Weil pairing ([4]) or the Tate pairing. The security of the schemes described in this paper relies on the hardness of the following problems.

Definition 1. *Given groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$,*

- *the **Bilinear Diffie-Hellman problem** (BDH) in ($\mathbb{G}_1$,$\mathbb{G}_2$,$\hat{e}$) is to compute $\hat{e}(P,P)^{abc}$ given (P, aP, bP, cP).*
- *The **Decisional Bilinear Diffie-Hellman problem** (DBDH) is, given (P, aP, bP, cP) and $z \in \mathbb{G}_2$, to decide whether $z = \hat{e}(P,P)^{abc}$ or not. The advantage of a distinguisher $\mathcal{D}$ for the DBDH problem is defined as*

$$Adv(\mathcal{D}) = \big|Pr_{a,b,c \in_R \mathbb{Z}_q, h \in_R \mathbb{G}_2}[1 \leftarrow \mathcal{D}(aP, bP, cP, h)] - Pr_{a,b,c \in_R \mathbb{Z}_q}[1 \leftarrow \mathcal{D}(aP, bP, cP, \hat{e}(P,P)^{abc})]\big|.$$

- *The **Gap Bilinear Diffie Hellman problem** is to solve a given instance (P, aP, bP, cP) of the BDH problem with the help of a DBDH oracle that is able to decide whether a tuple $(P, a'P, b'P, c'P, z)$ is such that $z = \hat{e}(P,P)^{a'b'c'}$ or not. Such tuples will be called DBDH tuples.*

The DBDH problem was introduced in [14] where it is shown to be not harder than the decisional Diffie-Hellman problem in $\mathbb{G}_2$. It is not known whether the Bilinear Diffie-Hellman problem is strictly easier than the computational Diffie-Hellman problem in $\mathbb{G}_1$ or not. It is also an open question whether the DBDH problem is strictly easier than the BDH one (although it is obviously not harder). Nevertheless, no probabilistic polynomial time algorithm is known to solve any of them with a non-negligible advantage so far. Their hardness then seems to be a reasonable assumption for the security of cryptographic protocols.

2.2 Identity based undeniable signatures

An identity based undeniable signature is made of three algorithms and two possibly interactive protocols.

Setup: the PKG takes as input a security parameter k and produces a public/private key pair (s, P_{pub}) and the system's public parameters `params`. s is the system's master key and P_{pub} is the PKG's public key that must be certified.

Keygen: given a user's identity ID, the PKG uses its master secret key s to compute the corresponding private key d_{ID} and transmit it to the user through a secure channel.

Sign: given a message $M \in \{0,1\}^*$ and his private key d_{ID}, the user generates a signature σ associated to M for his identity ID.

Confirm: is a protocol between a signer and a (possibly designated) verifier that takes as input a message $M \in \{0,1\}^*$, an identity $ID \in \{0,1\}^*$, the associated private key d_{ID} and a valid signature σ for the pair (M, ID). The output of the protocol is a (possibly non-interactive) non-transferable proof that σ is actually a valid signature on M for the identity ID.

Deny: is a protocol of the same kind as **Confirm** but its input is an invalid signature σ for a given pair (M, ID) and the private key d_{ID} corresponding to ID. Its output is a proof that σ is not a valid signature for a message M and an identity ID.

Confirm and Deny may be a single protocol. In our scheme, they are distinct.

2.3 Security notions for identity based undeniable signatures

The first security notion for ID-based undeniable signature is close to the one for other existing identity based signatures: it is the notion of existential unforgeability under chosen-message attacks.

Definition 2. *We say that an identity based undeniable signature scheme is **existentially unforgeable** under chosen-message attacks if no probabilistic polynomial time (PPT) adversary has a non-negligible advantage in the following game:*

1. *The challenger runs the setup algorithm to generate the system's parameters and sends them to the adversary.*
2. *The adversary $\mathcal{F}$ performs a series of queries:*
   - *Key extraction queries: $\mathcal{F}$ produces an identity ID and receives the private key d_{ID} corresponding to ID.*
   - *Signature queries: $\mathcal{F}$ produces a message M and an identity ID and receives a signature on M that was generated by the signature oracle using the private key corresponding to the public key ID.*
   - *Confirmation/denial queries: $\mathcal{F}$ produces a pair message-signature (M, σ) and an identity ID and gives them to the signature oracle that runs the confirmation/denial protocol to convince $\mathcal{F}$ that σ is actually related to M and ID or that it is not (in a non-transferable way) using the private key d_{ID} corresponding to ID.*
3. *After a polynomial number of queries, $\mathcal{F}$ produces a tuple (ID, M, σ) made of an identity ID, whose corresponding private key was not asked to the challenger during stage 2, and a message-signature pair (M, σ) that was not issued by the signature oracle during stage 2 for the identity ID.*

The forger $\mathcal{F}$ wins the game if it is able to provide a non-transferable proof of validity of the signature σ for message M and identity ID. Its advantage is defined to be its probability of success taken over the coin-flippings of the challenger and $\mathcal{F}$.

A second security notion for undeniable signatures was introduced by Chaum, van Heijst and Pfitzmann ([13]) and is called 'invisibility'. Informally, this notion corresponds to the inability for a dishonest verifier to decide whether a given signature on a given message was issued by some signer even after having observed several executions of confirmation/denial protocols by the same signer for other signatures. Galbraith and Mao ([21]) proposed a general definition for this security notion. In the identity based setting, we need to strengthen it a little to consider the fact that a dishonest user might be in possession of private keys associated to other identities before trying to validate or invalidate an alleged signature on a message for an identity without the help of the alleged signer.

Definition 3. *An identity based undeniable signature scheme is said to satisfy the **invisibility** property if no PPT distinguisher $\mathcal{D}$ has a non-negligible advantage against a challenger in the following game:*

1. *The challenger performs the setup of the scheme and sends the system's public parameters to $\mathcal{D}$.*
2. *The distinguisher $\mathcal{D}$ performs a polynomially bounded number of queries: key extraction queries, signature queries and confirmation/denial queries of the same kind as those of the previous definition.*
3. *After a first series of queries, $\mathcal{D}$ asks for a challenge: it produces a pair (M, ID) made of a message and an identity for which the associated private key was not asked in step 2. The challenger then flips a coin $b \leftarrow_R \{0, 1\}$. If $b = 0$, the challenger sends $\mathcal{D}$ a valid signature σ on M for the identity ID. Otherwise, $\mathcal{D}$ receives from the challenger a random element $\sigma \leftarrow_R \mathcal{S}$ taken at random from the signature space $\mathcal{S}$.*

4. The distinguisher $\mathcal{D}$ then performs a second series of queries. This time, it is not allowed to perform a confirmation/denial query for the challenge (σ, M, ID) nor to ask the private key associated to ID.
5. At the end of the game, $\mathcal{D}$ outputs a bit b' (that is 0 if $\mathcal{D}$ finds that (σ, M, ID) is a valid message-signature-identity tuple and 1 otherwise) and wins the game if $b = b'$.

$\mathcal{D}$'s advantage in this game is defined to be $Adv^{inv}(\mathcal{D}) := 2Pr[b = b'] - 1$.

The above probability is taken over the coin flippings of the distinguisher $\mathcal{D}$ and the challenger.

Similarly to what is done in [21], we also consider the notion of anonymity. This notion is slightly strengthened in the identity based setting

Definition 4. *We say that an identity based undeniable signature scheme satisfies the **anonymity** property if no probabilistic polynomial time distinguiser $\mathcal{D}$ has a non-negligible advantage in the following game:*

1. The challenger performs the setup of the scheme and sends the system's public parameters to $\mathcal{D}$.
2. The distinguisher $\mathcal{D}$ performs a polynomially bounded number of queries: key extraction queries, signature queries and confirmation/denial queries of the same kind as those of definition 2.
3. After a first series of queries, $\mathcal{D}$ requests a challenge: it produces a message M and a pair of identities ID_0, ID_1 for which it did not ask the associated private keys in the first stage. The challenger then flips a coin $b \leftarrow_R \{0, 1\}$ and computes the signature σ on M with the private key associated to ID_b. σ is sent as a challenge to $\mathcal{D}$.
4. $\mathcal{D}$ performs another series of queries. This time, it is not allowed to perform a confirmation or denial query for the challenge σ on identities ID_0, ID_1 nor to request the private key associated to these identities.

At the end of the game, $\mathcal{D}$ outputs a bit b' for which it finds that σ is a valid signature on M for the identity $ID_{b'}$. It wins the game if $b' = b$. Its advantage is defined as the previous definition.

It is shown in [21] that the notions of invisibility and anonymity are essentially equivalent for undeniable and confirmer signature schemes satisfying some particular properties. It is almost straightforward (by using the techniques of [21]) to show that this equivalence also holds in the identity based setting. We will not do it here. In the next section, we describe a first example of identity based undeniable signature and we just focus on proving its existential unforgeability and its invisibility in the random oracle model.

3 An identity based undeniable signature

Our ID-based undeniable signature scheme is made of the following five algorithms.

Setup: given security parameters k and l, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$, a generator P of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1, H_2 : \{0,1\}^* \times \{0,1\}^l \times \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_3, H_4 : \{0,1\}^* \rightarrow \mathbb{Z}_q$. It chooses a master secret $s \leftarrow_R \mathbb{Z}_q$ and computes $P_{pub} = sP \in \mathbb{G}_1$ that is made public. The system's public parameters are

$$\texttt{params} := \{q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4\}.$$

Keygen: given an identity ID, the PKG computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ and the associated private key $d_{ID} = sQ_{ID} \in \mathbb{G}_1$ that is transmitted to the user.

Sign: to sign a message $M \in \{0,1\}^*$, the signer Alice uses the private key d_{ID_A} associated to her identity ID_A.

1. She picks a random string $r \leftarrow_R \{0,1\}^l$ to compute $H_2(M, r, ID_A) \in \mathbb{G}_1$.
2. She then computes $\gamma = \hat{e}(H_2(M, r, ID_A), d_{ID_A}) \in \mathbb{G}_2$. The signature on M is given by

$$\sigma = (r, \gamma) = (r, \hat{e}(H_2(M, r, ID_A), d_{ID_A})) \in \{0,1\}^l \times \mathbb{G}_2.$$

Confirm: to verify a signature σ on a message M, a verifier of identity ID_B needs the help of the signer Alice. He sends her the pair (M, σ), where $\sigma =< r, \gamma >\in \{0,1\}^l \times \mathbb{G}_2$ is the alleged signature. Alice then runs the following confirmation protocol to produce a non-interactive designated-verifier proof that σ is a valid signature on M for her identity ID_A:

a. She first computes $Q_{ID_B} = H_1(ID_B)$.
b. She picks $U, R \leftarrow_R \mathbb{G}_1$ and $v \leftarrow_R \mathbb{Z}_q$ and computes

$$c = \hat{e}(P, U)\hat{e}(P_{pub}, Q_{ID_B})^v \in \mathbb{G}_2$$
$$g_1 = \hat{e}(P, R) \in \mathbb{G}_2 \text{ and } g_2 = \hat{e}(H_2(M, r, ID_A), R) \in \mathbb{G}_2.$$

c. She takes the hash value $h = H_3(c, g_1, g_2, M, r, \gamma) \in \mathbb{Z}_q$.
d. She computes $S = R - (h + v)d_{ID_A}$.

The proof is made of (U, v, h, S) and is checked by the verifier like this: he first computes $c' = \hat{e}(P, U)\hat{e}(P_{pub}, Q_{ID_B})^v$ and then $g_1' = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^{h+v}$ and $g_2' = \hat{e}(H_2(M, r, ID_A), S)\gamma^{h+v}$. He accepts the proof if and only if $h = H_3(c', g_1', g_2', M, r, \gamma)$.

Deny: in order to convince a designated verifier of identity ID_B that a given signature $\sigma =< r, \gamma >$ on a message M is not valid for her identity ID_A,

a. Alice computes $Q_{ID_B} = H_1(ID_B) \in \mathbb{G}_1$ and picks random $U \leftarrow_R \mathbb{G}_1$, $v \leftarrow_R \mathbb{Z}_q$ to compute $c = \hat{e}(P, U)\hat{e}(P_{pub}, Q_{ID_B})^v$.
b. She computes a commitment $C = \left(\frac{\hat{e}(H_2(M,r,ID_A), d_{ID_A})}{\gamma}\right)^\omega$ for a randomly chosen $\omega \leftarrow_R \mathbb{Z}_q$.
c. She proves in a zero-knowledge way that she knows a pair $(R, \alpha) \in \mathbb{G}_1 \times \mathbb{Z}_q$ such that

$$C = \frac{\hat{e}(H_2(M, r, ID_A), R)}{\gamma^\alpha} \quad \text{and} \quad 1 = \frac{\hat{e}(P, R)}{\hat{e}(P_{pub}, Q_{ID_A})^\alpha} \tag{1}$$

To do this,

1. She picks $V \leftarrow_R \mathbb{G}_1$, $v \leftarrow_R \mathbb{Z}_q$ to compute

$$\rho_1 = \hat{e}(H_2(M, r, ID_A), V)\gamma^{-v} \in \mathbb{G}_2 \text{ and } \rho_2 = \hat{e}(P, V)y^{-v} \in \mathbb{G}_2$$

where $y = \hat{e}(P_{pub}, Q_{ID_A})$.
2. She computes $h = H_4(C, c, \rho_1, \rho_2, M, r, \gamma) \in \mathbb{Z}_q$.
3. She computes $S = V + (h + v)R \in \mathbb{G}_1$ and $s = v + (h + v)\alpha \in \mathbb{Z}_q$.

The proof is made of (C, U, v, h, S, s). It can be verified by the verifier of identity ID_B who rejects the proof if $C = 1$ and otherwise computes $c' = \hat{e}(P, U)\hat{e}(P_{pub}, Q_{ID_B})^v$, $\rho_1' = \hat{e}(H_2(M, r, ID_A), S)\gamma^{-s}C^{-(h+v)}$ and $\rho_2' = \hat{e}(P, S)y^{-s}$ where $y = \hat{e}(P_{pub}, Q_{ID_A})$. The verifier accepts the proof if and only if $h = H_4(C, c', \rho_1', \rho_2', M, r, \gamma)$.

The confirmation protocol is a pairing based adaptation of a (repaired [3]) designated verifier proof ([27]) proposed by Jakobsson, Sako and Impagliazzo that allows a prover to convince a designated verifier of the equality of two discrete logarithms. The denial protocol is an adaptation of a protocol proposed by Camenisch and Shoup ([8]) to prove the inequality of two discrete logarithms. Both adaptations are non-transferable proofs of respectively equality and inequality of two inverses of the group isomorphisms $f_Q : \mathbb{G}_1 \rightarrow \mathbb{G}_2, Q \rightarrow f_Q(U) = \hat{e}(Q, U)$ with $Q = P$ and $Q = H_2(M, r, ID_A)$. In an execution of the confirmation protocol, the verifier B takes the signature as valid if he is convinced that $f_P(d_{ID_A}) = \hat{e}(P_{pub}, Q_{ID_A})$ and γ have identical pre-images for isomorphisms $f_P(.) = \hat{e}(P, .)$ and $f_{H_2(M,r,ID_A)}(.) = \hat{e}(H_2(M, r, ID_A), .)$. In the denial protocol, he takes the signature as invalid on M for identity ID_A if he is convinced that these inverses are different.

Completeness and soundness of the confirmation protocol: it is easy to see that a correct proof is always accepted by the verifier B: if (U, v, h, S) is correctly computed by the prover, we have $\hat{e}(P, S) = \hat{e}(P, R)\hat{e}(P, d_{ID_A})^{h+v}$ and $\hat{e}(P, d_{ID_A}) = \hat{e}(P_{pub}, Q_{ID_A})$. We also have $\hat{e}(H_2(M, r, ID_A), R) = \hat{e}(H_2(M, r, ID_A), S)\hat{e}(H_2(M, r, ID_A), d_{ID_A})^{h+v}$. In order to show the soundness, we notice that if a prover is able to provide two correct answers S_1, S_2 for the same commitment (c, g_1, g_2) and two different challenges h_1 and h_2, we then have the relations

$$\hat{e}(P, (h_1 - h_2)^{-1}(S_1 - S_2)) = \hat{e}(P_{pub}, Q_{ID_A})$$
$$\hat{e}(H_2(M, r, ID_A), (h_1 - h_2)^{-1}(S_1 - S_2)) = \gamma.$$

This shows that both inverses of $f_P^{-1}(\hat{e}(P_{pub}, Q_{ID_A}))$ and $f_{H_2(M,r,ID_A)}^{-1}(\gamma)$ are equal.

Completeness and soundness of the denial protocol: one easily checks that a honest prover is always accepted by the designated verifier. To prove the soundness, one notices that if the prover is able to provide a proof of knowledge of a pair (R, α) satisfying equations (1), then the second of these equations implies $R = \alpha f_P^{-1}(y)$ with $y = \hat{e}(P_{pub}, Q_{ID_A})$ by the bilinearity of the map. If we substitute this relation into the first equation of (1), it comes that

$$C = \Big(\frac{\hat{e}(H_2(M, r, ID_A), f_P^{-1}(y))}{\gamma}\Big)^{\alpha}.$$

Since the verifier checks that $C \neq 1$, it comes that $\hat{e}(H_2(M, r, ID_A), f_P^{-1}(y)) \neq \gamma$ and the signature γ is actually invalid. The soundness of the proof of knowledge in step c is easy to verify.

Non-transferability: in order for the proofs to be non-transferable, both protocols need a trapdoor commitment $\texttt{Commit}(U, v) = \hat{e}(P, U)\hat{e}(P_{pub}, Q_{ID_B})^v$ that allows the owner of the private key d_{ID_B} to compute commitment collisions: indeed, given a tuple $(U, v, \texttt{Commit}(U, v))$, B can easily use d_{ID_B} to find a pair (U', v') such that $\texttt{Commit}(U, v) = \texttt{Commit}(U', v')$. This is essential for the proof to be non-transferable: the verifier B cannot convince a third party of the validity or of the invalidity of a signature since his knowledge of the private key d_{ID_B} allows him to produce such a proof himself. Indeed, given a message-signature pair (M, σ), with $\sigma =< r, \gamma >\in \{0, 1\}^l \times \mathbb{G}_2$, B can choose $S \leftarrow_R \mathbb{G}_1$, $x \leftarrow_R \mathbb{Z}_q$ and $U' \leftarrow_R \mathbb{G}_1$ to compute $c = \hat{e}(P, U')$, $g_1 = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^x$, $g_2 = \hat{e}(H_2(M, r, ID_A), S)\gamma^x$ and $c = H_3(c, g_1, g_2, M, r, \gamma)$. He can then compute $v = x - h \bmod q$ and $U = U' - vd_{ID_B} \in \mathbb{G}_1$ where d_{ID_B} is the verifier's private key. (U, v, h, S) is thus a valid proof built by the verifier with the trapdoor d_{ID_B}. This trapdoor also allows him to produce a false proof of a given signature's invalidity using the same technique with the denial protocol.

[3] The original proof proposed by Jakobsson et al. suffer from a security flaw as well as the quite similar one used by Galbraith and Mao. This was noticed by Fangguo Zhang who also found how to easily fix these problems.

Efficiency considerations: From an efficiency point of view, the signature generation algorithm requires one pairing evaluation as a most expensive operation. The confirmation and denial protocols are more expensive: the first one requires 4 pairing evaluations (3 if $\hat{e}(P_{pub}, Q_{ID_B})$ is cached in memory: this can be done if the verifier often performs verification queries), one exponentiation in $\mathbb{G}_2$ and one computation of the type $\lambda_1 P + \lambda_2 Q$ in $\mathbb{G}_1$. The verifier needs to compute 3 pairings (2 if $\hat{e}(P_{pub}, Q_{ID_A})$ is cached), 3 exponentiations and 3 multiplications in $\mathbb{G}_2$. In the denial protocol, the prover must compute 5 pairings (4 if $\hat{e}(P_{pub}, Q_{ID_B})$ is cached), 4 exponentiations and 4 multiplications in $\mathbb{G}_2$, one computation of the type $\lambda_1 P + \lambda_2 Q$ and some extra arithmetic operations in $\mathbb{Z}_q$. The verifier must compute 4 pairings (3 if $\hat{e}(P_{pub}, Q_{ID_B})$ is cached), 2 exponentiations, 1 multi-exponentiation and 3 multiplications in $\mathbb{G}_2$. To improve the efficiency of the confirmation and denial algorithms, one can speed up the computation of commitments. Indeed, the prover can pre-compute $\hat{e}(P, P)$ once and for all. To generate a commitment in an execution of the confirmation protocol, he then picks $u, v, x \leftarrow_R \mathbb{Z}_q$ and computes $c = \hat{e}(P, P)^u \hat{e}(P_{pub}, Q_{ID_B})^v$, $R = xP$ $g_1 = \hat{e}(P, P)^x$, $g_2 = \hat{e}(H_2(M, r, ID_A), R)$. The answer to the challenge h must then be computed as $S = R + (h + v)d_{ID_A}$ and the proof is made of (u, v, h, S). This technique can also be applied in the denial protocol. It allows replacing 2 pairing evaluations by 2 scalar multiplications, one exponentiation and a multi-exponentiation in $\mathbb{G}_2$ (to compute c). A single pairing evaluation is then required for the prover at each execution of the confirmation and denial protocols.

Globally, it turns out that a signature verification is more expensive than a signature generation even if a pre-computation is performed. Our ID-based undeniable signature solution is nevertheless reasonable.

If we consider the length of signatures, the binary representation of a pairing is about 1000 bits long (1024 if we use the same curve as in [4]) while the length l of the binary string can be of the order of 100 bits. This provides us with signatures of about 1100 bits. This is roughly one half of the size of the RSA-based undeniable signature proposed in [21] (this scheme produces signatures of more than 2048 bits if 1024-bit moduli are used). If we compare our scheme with the original undeniable signature proposed by Chaum and van Heijst and proven secure by Okamoto and Pointcheval ([32]), both lengths are similar if the Chaum-van Heijst scheme is used over a group like $\mathbb{Z}_p$ with $|p| = 1000$ (this is no longer true if this scheme is used over a suitable elliptic curve). However, it remains an open problem to devise identity based undeniable signature schemes with shorter signatures than ours.

Convertible signatures It is really easy to notice that issued signatures can be selectively turned into universally verifiable signatures by the signer. In order to convert a genuine signature $\sigma =< r, \hat{e}(H_2(M, r, ID_A), d_{ID_A}) >$, the signer Alice just has to take a random $x \leftarrow_R \mathbb{Z}_q$ and compute $R = xP$, $g_1 = \hat{e}(P, P)^x$, $g_2 = \hat{e}(H_2(M, r, ID_A), R)$, the hash value $h = H(g_1, g_2, M, r, \gamma)$ and the answer $S = R + hd_{ID_A}$. The proof, given by $(h, S) \in \mathbb{Z}_q \times \mathbb{G}_1$, is easily universally verifiable by a method similar to the verification in the confirmation protocol. Alice can also give a universally verifiable proof that a given signature is invalid for her identity by using the non-designated verifier counterpart of the denial protocol.

Removing key escrow In order to prevent a dishonest PKG from issuing a signature on behalf of a user and from compromising the invisibility and anonymity properties, one can easily use the generic transformation proposed by Al-Riyami and Paterson ([1]) to turn the scheme into a certificateless undeniable signature. The obtained scheme is provably secure in the model described in [1] in the same way as its identity based counterpart in the identity based security model. Unfortunately, the advantage of easy key management is lost since the resulting scheme no longer supports human-memorizable public keys. On the other hand, key escrow, which is often an undesirable feature in signature schemes, is then removed as well as the need for public key certificates.

4 Security proofs for the ID-based undeniable signature

We first give a proof in the random oracle model that our identity based undeniable signature is existentially unforgeable under adaptive chosen-message attacks. We then provide a proof of its invisibility.

Theorem 1. *In the random oracle model, if there exists an adversary $\mathcal{F}$ that is able to succeed in an existential forgery against the identity based undeniable signature scheme described in the previous section with an advantage ϵ within a time t and when performing q_E key extraction queries, q_S signature queries, q_{CD} confirmation/denial queries and q_{H_i} queries on hash oracles H_i, for $i = 1, \ldots, 4$, then there exists an algorithm $\mathcal{B}$ that is able to solve the Bilinear Diffie-Hellman problem with an advantage*

$$\epsilon' \geq \frac{\epsilon - (2q_{H_3} + q_{CD} + 1)/2^k}{e^2(q_E + 1)(q_{CD} + 1)}$$

in a time $t' \leq t + 6\mathcal{T}_p + (q_E + q_{H_1} + q_{H_2})\mathcal{T}_m + (q_S + q_{CD})\mathcal{T}_e + 2q_{CD}\mathcal{T}_{me}$ where $\mathcal{T}_p$ denotes the time required for a pairing evaluation, $\mathcal{T}_m$ is the time to perform a multiplication in $\mathbb{G}_1$, $\mathcal{T}_e$ is the time to perform an exponentiation in $\mathbb{G}_2$, $\mathcal{T}_{me}$ the time for a multi-exponentiation in $\mathbb{G}_2$ and e is the base for the natural logarithm.

Proof. see the appendix $\square$

In order for the proof to hold, we must have $q_{H_2} \ll 2^l$, where l is the size of the random salt r. We then take $l = 100$. We note that the reduction is not really efficient: if we take $q_E \approx q_{CD} \leq 2^{30}$ and $q_{H_3} < 2^{80}$, we then have $(2q_{H_3} + q_{CD} + 1)2^{-k} \approx 1.65 \times 10^{-64}$. If we assume $\epsilon - (2q_{H_3} + q_{CD} + 1)2^{-k} > \epsilon/2$, we obtain the bound $\epsilon' > \epsilon/2^{64}$. However, we have a proof with a tighter bound if the underlying assumption is the hardness of the Gap Bilinear Diffie-Hellman problem. The use of a DBDH oracle allows algorithm $\mathcal{B}$ to perfectly simulate the confirmation/denial protocols. The advantage of algorithm $\mathcal{B}$ is then

$$\epsilon' \geq \frac{\epsilon - (2q_{H_3} + q_{CD} + 1)2^{-k}}{e(q_E + 1)} > \epsilon/2^{32}.$$

We note that using the techniques of Katz and Wang ([28]) easily allows replacing the random salt r by a single bit and then obtaining signatures that are about 100 bits shorter without losing security guarantees.

The theorem below claims the scheme's invisibility in the sense of Galbraith and Mao (see [21]) under the Decisional Bilinear Diffie-Hellman assumption.

Theorem 2. *In the random oracle model, the identity based undeniable signature presented in section 3 satisfies the invisibility property provided the Decisional Bilinear Diffie-Hellman problem is hard. More formally, if we assume that no algorithm is able to forge a signature in the game of definition 2 with a non-negligible probability and if a distinguisher $\mathcal{D}$ is able to distinguish valid signature from invalid ones for a messages and an identity of its choice with a non-negligible advantage ϵ after having asked q_E key extraction queries, then there exists a distinguisher $\mathcal{B}$ that has an advantage $\epsilon' \geq \frac{\epsilon}{e(q_E+1)}$ for the DBDH problem within a time bounded as in theorem 1.*

Proof. see the appendix $\square$

It is possible to directly show that the scheme also satisfies the anonymity property in the random oracle model under the Decisional Bilinear Diffie-Hellman assumption. However, since anonymity and invisibility are essentially equivalent, the anonymity of our signature derives from its invisibility property.

5 Another application

Our construction provides an application of independent interest which is the possibility to design an identity based signature with a 'tighter' security proof than all other existing provably secure identity based signatures ([9],[18],[24],[26]) for which the security proofs make use of the forking lemma ([35],[36]): indeed by concatenating a signature produced by the undeniable scheme with a non-designated verifier proof of its validity (using the non-designated verifier counterpart of the confirmation protocol), we obtain a universally verifiable identity based signature for which the security is the most tightly related to some hard computational problem. Recall that all existing identity based signatures have a security proof built on the forking lemma of Pointcheval and Stern that involves a degradation in security during the reduction as pointed out in [23]: if q_H denotes the number of message hash queries and t the forger's running time, then the upper bound on the average running time to solve the problem is $q_H t$ (if we assume $q_H < 2^{80}$, this makes a great degradation for the bound on the running time). This new ID-based signature may be viewed as an adaptation of the signature recently proposed by Goh and Jarecki ([23]). It is less efficient than all the other ones but is more tightly related to a computational assumption than those in ([9],[26]), which are only loosely related to the computational Diffie-Hellman problem, or the scheme in ([24]) that has a security loosely related to the RSA assumption. As for the proof of unforgeability of the undeniable signature under the Gap Bilinear Diffie-Hellman assumption, one can show that, if the forger's advantage is ϵ, then the average time to solve the BDH problem is smaller than $2^{32}t/\epsilon$ where t is the running time of the forger. The corresponding bound for the identity based signature described in [26] is roughly $2^{146}t/\epsilon$ if 2^{30} identity hash queries and 2^{80} message hash queries [4] are allowed to the attacker.

6 Conclusions

In this paper, we showed a first construction for a provably secure identity based undeniable signature and we extended the panel of primitives for identity based cryptography ([1]). We provided a proof of existential unforgeability under the Bilinear Diffie-Hellman assumption and we argued that a 'tighter' proof can be made under the Gap Bilinear Diffie-Hellman assumption. Our construction and the underlying assumption for its security are inspired from those of Chaum-van Antwerpen ([11]) and Okamoto-Pointcheval ([32]). We also extended the notions of invisibility and anonymity of Galbraith and Mao ([21]) to the identity based setting and we proved the invisibility of our scheme in the random oracle model under the Decisional Bilinear Diffie-Hellman assumption.

As a side effect, our construction allows the design an identity based signature scheme with a security more tightly related to the hardness of some hard problem than any other existing provably secure identity based signature.

A direction for future research would be to find an identity based undeniable signature scheme that satisfies the invisibility property and is tightly related to a weaker assumption than the hardness of the BDH problem.

Acknowledgements

We wish to thank Fangguo Zhang for his suggestion to correct the confirmation protocol.

References

1. S.-S. Al-Riyami , K.G. Paterson, *Certificateless Public Key Cryptography*, Advances in Cryptology - Asiacrypt'03, Lecture Notes in Computer Science Series, 2003.

[4] Hashing onto a finite field may be viewed as an operation of unit cost while hashing onto an elliptic curve requires some extra computation as explained in [2].

2. P.-S.-L.-M. Barreto, H.-Y. Kim, *Fast hashing onto elliptic curves over fields of characteristic 3*, eprint available at http://eprint.iacr.org/2001/098/.
3. M. Bellare, P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, Proc. of the 1^{st} ACM Conference on Computer and Communications Security, pp. 62-73, 1993.
4. D. Boneh, M. Franklin, *Identity Based Encryption From the Weil Pairing*, Advances in Cryptology - Crypto'01, Lecture Notes in Computer Science vol. 2139, Springer-Verlag, pp. 213-229, 2001.
5. D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, Advances in Cryptology - Asiacrypt'01, Lecture Notes in Computer Science vol. 2248, Springer, pp. 514-532, 2001.
6. J. Boyar, D. Chaum, I. Damgård, T. Pedersen, *Convertible undeniable signatures*, Advances in Cryptology - Crypto'90, Lecture Notes in Computer Science vol. 537, Springer, pp. 189-208, 1990.
7. X. Boyen, *Multipurpose Identity-Based Signcryption. A Swiss Army Knife for Identity-Based Cryptography*, Advances in Cryptology - Crypto'03, Lecture Notes in Computer Science vol. 2729, Springer, pp. 383-399, 2003.
8. J. Camenisch, V. Shoup, *Practical Verifiable Encryption and Decryption of Discrete Logarithms*, Advances in Cryptology - Crypto'03, Lecture Notes in Computer Science vol. 2729, Springer, pp. 126-144, 2003.
9. J.C. Cha, J.H. Cheon, *An Identity-Based Signature from Gap Diffie-Hellman Groups*, proceedings of PKC 2003. Springer Verlag, Lecture Notes in Computer Science vol. 2567, pp. 18-30, Springer-Verlag, 2003.
10. D. Chaum, T. Pedersen, *Wallet databases with observers*, Advances in Cryptology - Crypto'92, Lecture Notes in Computer Science vol. 740, pp. 89-105, Springer-Verlag, 1992.
11. D. Chaum, H. van Antwerpen, *Undeniable signatures*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Science vol. 435, Springer-Verlag, pp. 212-216, 1989.
12. D. Chaum, *Zero-knowledge undeniable signatures*, Advances in Cryptology - Crypto'90, Lecture Notes in Computer Science vol. 473, Springer-Verlag, pp. 458-464, 1990.
13. D. Chaum, E. van Heijst, B. Pfitzmann, *Cryptographically strong undeniable signatures, unconditionally secure for the signer*, Advances in Cryptology - Crypto'91, Lecture Notes in Computer Science vol. 576, Springer-Verlag, pp. 470-484, 1991.
14. J.H. Cheon, D. H. Lee, *Diffie-Hellman Problems and Bilinear Maps*, eprint available at http://eprint.iacr.org/2002/117/.
15. C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Proc. of Cryptography and Coding, Lecture Notes in Computer Science 2260, Springer, pp. 360-363, 2001.
16. R. Dupont, A. Enge, *Practical Non-Interactive Key Distribution Based on Pairings*, available at http://eprint.iacr.org/2002/136.
17. I. Damgård, T. Pedersen, *New convertible undeniable signature schemes*, Advances in Cryptology - Eurocrypt'96, Lecture Notes in Computer Science vol. 1070, pp. 372-386, Springer-Verlag, 1996.
18. A. Fiat, A. Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems* , Advances in Cryptology - Crypto'86, Lecture Notes in Computer Science 0263, Springer-Verlag, pp. 186-194, 1986.
19. A. Fujioka, T. Okamoto, K. Ohta, *Interactive Bi-Proof Systems and undeniable signature schemes*, Advances in Cryptology - Eurocrypt'91, Lecture Notes in Computer Science vol. 547, pp. 243-256, Springer-Verlag, 1991.
20. S. Galbraith, W. Mao, K.G. Paterson, *RSA-based undeniable signatures for general moduli*, Topics in Cryptology - CT-RSA 2002. Lecture Notes in Computer Science vol. 2271, Springer Verlag, pp. 200-217.
21. S. Galbraith, W. Mao, *Invisibility and Anonymity of Undeniable and Confirmer Signatures.*, proceedings of CT-RSA 2003. Springer Verlag, Lecture Notes in Computer Science series.
22. R. Gennaro, H. Krawczyk, T. Rabin, *RSA-based undeniable signatures*, Advances in Cryptology - Crypto'97, Lecture Notes in Computer Science vol. 1294, Springer-Verlag, pp. 132-149, 1997.
23. E.-J. Goh, S. Jarecki, *A Signature Scheme as Secure as the Diffie-Hellman Problem*, Advances in Cryptology - Eurocrypt'03, Lecture Notes in Computer Science vol. 2656, Springer-Verlag, pp. 401-415, 2003.
24. L. Guillou, J-J. Quisquater, *A "Paradoxical" Identity-Based Signature Scheme Resulting From Zero-Knowledge* , Advances in Cryptology - Crypto'88, Lecture Notes in Computer Science vol. 403, Springer-Verlag, pp. 216-231, 1988.
25. S. Han, K.Y. Yeung, J. Wang, *Identity based confirmer signatures from pairings over elliptic curves*, proceedings of ACM conference on Electronic commerce, pp. 262-263, 2003.
26. F. Hess, *Efficient identity based signature schemes based on pairings*, proceedings of SAC'02. Springer Verlag, Lecture Notes in Computer Science series.
27. M. Jakobsson, K. Sako, R. Impagliazzo, *Designated Verifier Proofs and Their Applications*, Advances in Cryptology - Eurocrypt'96, Lecture Notes in Computer Science vol. 1070, Springer-Verlag, pp. 143-154, 1996.
28. J. Katz, N. Wang, *Efficiency Improvements for Signature Schemes with Tight Security Reductions*, to appear at ACM Conference on Computer and Communications Security 2003.
29. A.J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 2^{nd} printing, 1995.
30. M. Michels, M. Stadler, *Efficient Convertible Undeniable Signature Schemes.* Proceedings of SAC'97
31. T. Okamoto, *Designated Confirmer Signatures and Public Key Encryption are Equivalent*, Advances in Cryptology - Crypto'94, Lecture Notes in Computer Science vol. 0839, Springer-Verlag, pp. 61-74, 1994.
32. T. Okamoto, D. Pointcheval *The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes*, Proc. of of PKC'01, LNCS 1992, Springer, pp.104-118, 2001.

33. K.G. Paterson, *ID-based signatures from pairings on elliptic curves*, available at http://eprint.iacr.org/2002/004/.
34. D. Pointcheval, *Self-Scrambling Anonymizers*, Proceedings of Financial Cryptography 2002, Lecture Notes in Computer Science vol. 1962, Springer-Verlag, pp. 259-275, 2001.
35. D. Pointcheval, J. Stern, *Security proofs for signature schemes*, Advances in Cryptology - Eurocrypt'96, Lecture Notes in Computer Science vol. 1070, Springer-Verlag, pp. 387-398, 1996.
36. D. Pointcheval, J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptology, vol. 13-Number 3, pp. 361-396, 2000.
37. R. Sakai, K. Ohgishi, M. Kasahara, *Cryptosystems based on pairing*, In The 2000 Sympoium on Cryptography and Information Security, Okinawa, Japan, January 2000.
38. A. Shamir, *Identity Based Cryptosystems and Signature Schemes*, Advances in Cryptology - Crypto' 84, Lecture Notes in Computer Science 0196, Springer, 1984.
39. N.P. Smart, *An identity based authenticated key agreement protocol based on the Weil pairing*, Electronic Letters, 38(13): 630-632, 2002.
40. F. Zhang, K. Kim, *ID-Based Blind Signature and Ring Signature from Pairings.* Advances in Cryptology - Asiacrypt'02, Lecture Notes in Computer Science vol. 2501, Springer-Verlag, 2002.
41. F. Zhang, R. Safavi-Naini, W. Susilo, *Attack on Han et al.'s ID-based Confirmer (Undeniable) Signature at ACM-EC'03*, eprint available at http://eprint.iacr.org/2003/129/

Appendix

Proof of theorem 1

We show that one can build an algorithm $\mathcal{B}$ that can solve a random instance (P, aP, bP, cP) of the Bilinear Diffie-Hellman problem using the adversary $\mathcal{F}$. $\mathcal{B}$ will play the role of $\mathcal{F}$'s challenger in the game described in section 3. It first provides $\mathcal{F}$ with system parameters $\texttt{params} = \{q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4\}$ such that $P_{pub} = cP$ (where c is unknown to $\mathcal{B}$) and where H_1, H_2, H_3 and H_4 are random oracles.

$\mathcal{F}$ now performs a series of queries as described in defintion 2.3. Let us see how $\mathcal{B}$ can simulate the oracles that answer to these queries. It uses a list L_{H_1} to keep track of answers to hash queries on H_1, a list L_{H_2} for hash queries on H_2 and lists L_{H_3} and L_{H_4} for queries on H_3 and H_4. Without loss of generality, we can assume that hash queries on H_1 are distinct and that every key extraction on an identity ID is preceded by a random oracle query $H_1(ID)$ on that identity. The queries made by $\mathcal{F}$ are handled like this:

- queries on oracle H_1: to handle such a query on an identity $ID_i \in \{0,1\}^*$, $\mathcal{B}$ first picks $\mu_i \leftarrow_R \mathbb{Z}_q$. It then flips a coin X that is a random variable taking the value 0 with probability δ_1 and the value 1 with probability $1 - \delta_1$ (the optimal value of δ_1 will be determined further). $\mathcal{B}$ then inserts the tuple (ID_i, μ_i, X) into the list L_{H_1}. If $X = 1$, $\mathcal{B}$ returns $H_1(ID_i) = \mu_i(bP) \in \mathbb{G}_1$ to $\mathcal{F}$ (recall that b is unknown to $\mathcal{B}$). Otherwise, $\mathcal{B}$ returns $H_1(ID_i) = \mu_i P \in \mathbb{G}_1$ as an answer to $\mathcal{F}$.
- queries on oracle H_2: when receiving a query $H_2(M_i, r_i, ID_i)$, $\mathcal{B}$ first scans L_{H_2} to check if it contains a tuple (M_i, r_i, ID_i, d, Y). In this case, if $Y = 1$, $\mathcal{B}$ then answers $H_2(M_i, r_i, ID_i) = d(aP)$ to $\mathcal{F}$. If $Y = 0$, $\mathcal{B}$ then answers $H_2(M_i, r_i, ID_i) = dP$. If no tuple $(M_i, r_i, ID_i, ., .)$ is found in L_{H_2}, then $\mathcal{B}$ picks a random $d_i \leftarrow_R \mathbb{Z}_q$ and flips a coin Y that is a random variable taking the value 0 with probability δ_2 and the value 1 with probability $1 - \delta_2$. $\mathcal{B}$ then inserts the tuple (M_i, r_i, ID_i, d_i, Y) into L_{H_2}. If $Y = 1$, $\mathcal{B}$ returns $H_2(M_i, r_i, ID_i) = d_i(aP)$ (where $aP \in \mathbb{G}_1$ is unknown to $\mathcal{B}$) to $\mathcal{F}$. Otherwise, it returns $H_2(M_i, r_i, ID_i) = d_i P$
- key extraction queries: when $\mathcal{F}$ asks the private key associated to an identity ID_i, $\mathcal{B}$ scans L_{H_1} for a tuple (ID_i, μ, X) (such a tuple must be in L_{H_1} because of the assumption made above). If $X = 1$, then $\mathcal{B}$ outputs "failure" and stops since it is unable to answer the query. Otherwise, $\mathcal{B}$ returns μP_{pub} to $\mathcal{F}$ as the private key associated to ID_i.
- signature queries: for a signature query on a message M_i and an identity ID_i, $\mathcal{B}$ first chooses a random binary string $r \leftarrow_R \{0,1\}^l$ and checks if L_{H_2} already contains a tuple $(M_i, r, ID_i, ., .)$. If it does, $\mathcal{B}$ picks another random bitstring $r \leftarrow_R \{0,1\}^l$ until finding one for which no tuple $(M_i, r, ID_i, ., .)$ exists in L_{H_2}. Once it has an admissible r, $\mathcal{B}$ takes a random $d \leftarrow_R \mathbb{Z}_q$ and inserts $(M_i, r, ID_i, d, 0)$ into L_{H_2} (in such a way that a subsequent $H_2(M_i, r, ID_i)$ query will receive dP as an answer). Because of the assumptions made above, L_{H_1} must contain an entry $(ID_i, ., .)$ indicating what value was returned to $\mathcal{F}$ on its previously issued $H_1(ID_i)$ query. Let Q_{ID_i} the answer given by $\mathcal{B}$ to this query (this answer is easily recovered from L_{H_1} by $\mathcal{B}$). $\mathcal{B}$ then returns $\sigma =< r, \hat{e}(dP_{pub}, Q_{ID_i}) >$ as an answer to $\mathcal{F}$'s signature request.
- confirmation/denial queries: at any time, $\mathcal{F}$ can produce a message-signature pair (M, σ) and identities ID_A (the one of the alleged signer) and ID_B (the one of the designated verifier) and request an execution of the confirmation/denial protocol. To handle such a query, $\mathcal{B}$ first parses σ into $(r, \gamma) \in \{0,1\}^l \times \mathbb{G}_2$. The success of the simulation depends on whether L_{H_2} contains an entry $(M, r, ID_A, ., .)$ or not. We can distinguish three cases. First, if no such query is found, $\mathcal{B}$ can proceed as in the signature oracle simulation to generate a valid signature (r, γ') for the identity ID_A. After that, if $\gamma' = \gamma$, $\mathcal{B}$ simulates the confirmation protocol to convince $\mathcal{F}$ of the signature's validity. If $\gamma \neq \gamma'$, it simulates the denial protocol to convince $\mathcal{F}$ of the signature's invalidity.

A second possible case is that L_{H_2} already contains an entry $(M, r, ID_A, d, 0)$ for some $d \in \mathbb{Z}_q$ (that means that a previous query $H_2(M, r, ID_A)$ received dP as an answer). $\mathcal{B}$ can then recover $Q_{ID_A} = H_1(ID_A)$ from L_{H_1} and check whether $\gamma = \hat{e}(dP_{pub}, Q_{ID_A})$. $\mathcal{B}$ simulates the confirmation protocol if the latter condition holds and the denial protocol otherwise.

The last possible case is the one where L_{H_2} already contains a tuple $(M, r, ID_A, d, 1)$ for some $d \in \mathbb{Z}_q$ (indicating the $H_2(M, r, ID_A)$ was answered to be $d(aP)$). In this case, $\mathcal{B}$ scans L_{H_1} to find a tuple (ID_A, μ, X) (such a tuple must exist because of the assumptions made above). If $X = 1$, $\mathcal{B}$ outputs "failure" and stops since it is unable to compare γ with the legitimate signature on M with the random string r for the identity ID_A. If $X = 0$, $\mathcal{B}$ knows that $H_1(ID_A)$ was defined to be $\mu P \in \mathbb{G}_1$ (and that the associated private key should be μP_{pub}) and this allows it to compare the γ of the alleged signature with $\hat{e}(d(aP), \mu P_{pub})$. If both quantities are equal, $\mathcal{B}$ simulates the validation of the alleged signature σ and its invalidation otherwise.

The non-interactive proofs provided by the confirmation and denial protocols are very easy to simulate in the random oracle model. We do not give the details here but we notice that $\mathcal{B}$ can fail to simulate the confirmation and denial protocols with a probability smaller than $q_{H_3}2^{-k}$ (we assume $q_{H_3} \approx q_{H_4}$). This occurs if $\mathcal{B}$ has to fix the value of H_3 or H_4 on a point where the oracle was previously defined.

At the end of the game, $\mathcal{F}$ produces a tuple (M, ID, σ) where $\sigma =< r, \gamma > \in \{0,1\}^* \times \mathbb{G}_2$ is an alleged signature of a signer of identity ID on the message M. To win the game, $\mathcal{F}$ must not have queried the private key associated to ID and it must be able to produce a non-transferable proof of validity of σ. For the outputted tuple, $\mathcal{B}$ scans L_{H_1} and L_{H_2} for entries (ID, μ, X) (such an entry must exist because of the assumptions made above) and (M, r, ID, d, Y). If $X = 0$ or $Y = 0$, then $\mathcal{B}$ outputs "failure" and stops. If no entry (M, r, ID, d, Y) exists in L_{H_2}, then $\mathcal{B}$ also fails. Otherwise, if (r, γ) is a valid signature on M for the identity ID and if both inverses $f_P^{-1}(\hat{e}(P_{pub}, Q_{ID_A}))$ and $f^{-1}_{H_2(M,r,ID)}(\gamma)$ are actually equal, then $\mathcal{B}$ can compute $\gamma^{\frac{1}{\mu d}}$ which is equal to the solution $\hat{e}(P, P)^{abc}$ of the Bilinear Diffie-Hellman problem (P, aP, bP, cP).

We still have to assess $\mathcal{B}$'s probability of success. The first way for $\mathcal{B}$ to reach a failure state is to receive a key extraction query on an identity ID_i for which it fixed $H_1(ID_i) = \mu_i(bP)$ for some $\mu_i \in \mathbb{Z}_q$. A failure of $\mathcal{B}$ can also occur when $\mathcal{F}$ performs a confirmation/denial query $(M, ID, < r, \gamma >)$ for which $H_2(M, r, ID)$ was defined to be $d(aP)$, for some $d \in \mathbb{Z}_q$, and $H_1(ID)$ to $\mu(bP)$ for some $\mu \in \mathbb{Z}_q$. $\mathcal{B}$ also fails if the forgery $(M, ID, < r, \gamma >)$ produced by $\mathcal{F}$ is such that $H_1(ID)$ was set to μP for some $\mu \in \mathbb{Z}_q$ or $H_2(M, r, ID)$ was defined as dP for some $d \in \mathbb{Z}_q$. If q_E denotes the number of key extraction queries and q_{CD} the number of confirmation/denial queries made by $\mathcal{F}$, it easily comes that the probability for $\mathcal{B}$ to avoid these failure cases is at least $\delta_1^{q_E}(1 - \delta_1)\delta_2^{q_{CD}}(1 - \delta_2)$. If $\mathcal{B}$ uses the optimal values $\delta_{1,opt} = q_E/(q_E + 1)$ and $\delta_{2,opt} = q_{CD}/(q_{CD} + 1)$, this probability is greater than $\frac{1}{e^2(q_E+1)(q_{CD}+1)}$. It is also possible that $\mathcal{F}$ does not query $H_2(M, r, ID)$, where (M, r, ID) is a part of its forgery, during the simulation. One easily sees that the probability for this to happen is smaller than $1/2^k$. Finally, the attacker $\mathcal{F}$ can also produce a forgery $(M, ID, < r, \gamma >)$ for which it is able to give a valid proof of validity but for which $y = \hat{e}(P_{pub}, Q_{ID}) = \hat{e}(P, d_{ID})$, $\gamma = \hat{e}(H_2(M, r, ID), d'_{ID})$ with $d_{ID} \neq d_{ID'}$. Since $\mathcal{F}$ can provide a proof (U, v, h, S) that $(M, ID, < r, \gamma >)$ is valid, we have $g_1 = \hat{e}(P, R) = \hat{e}(P, S)\hat{e}(P, d_{ID})^{h+v}$, $g_2 = \hat{e}(H_2(M, r, ID), R') = \hat{e}(H_2(M, r, ID), S)\hat{e}(H_2(M, r, ID), d'_{ID})^{h+v}$ and then $h = \log_{d_{ID}-d'_{ID}}(R - R') - v$. Such a case can only happen if a hash value $H_3(c, \hat{e}(P, R), \hat{e}(H_2(M, r), R'))$ is defined to be $\log_{d_{ID}-d'_{ID}}(R - R') - v$ by $\mathcal{B}$. The probability for this to happen is not greater than $q_{H_3}2^{-k}$. Finally, the probability for $\mathcal{B}$ to fail in the simulation of a confirmation/denial is less than $(q_{H_3} + q_{CD})2^{-k}$ (since L_{H_3} contains at most $q_{H_3} + q_{CD}$ entries). This gives us the announced bound

$$\frac{\epsilon - (2q_{H_3} + q_{CD} + 1)/2^k}{e^2(q_E + 1)(q_{CD} + 1)}.$$

The bound on the computation time derives from the fact that every request on H_1, H_2 and every signing request or key extraction request requires $\mathcal{B}$ to compute a scalar multiplication in $\mathbb{G}_1$. To handle confirmation/denial and signature queries, $\mathcal{B}$ can avoid pairing evaluations by pre-computing $\hat{e}(P, aP)$, $\hat{e}(P, bP)$, $\hat{e}(P, cP)$, $\hat{e}(aP, cP)$, $\hat{e}(aP, bP)$ and $\hat{e}(bP, cP)$ and performing exponentiations in $\mathbb{G}_2$. Every signature query requires thus an exponentiation in $\mathbb{G}_2$ while each confirmation/denial queries requires one exponentiation to be able to check the validity of the alleged signature and 2 multi-exponentiation in $\mathbb{G}_2$ to simulate the confirmation/denial protocol. This gives us the bound on $\mathcal{B}$'s running time.

$\square$

Proof of theorem 2

We assume there exists a distinguisher $\mathcal{D}$ that is able to decide whether a signature on a message was actually issued by a signer without the help of the latter. We show that such a distinguisher allows building a probabilistic polynomial time algorithm $\mathcal{B}$ that is able to solve the Decisional Bilinear Diffie-Hellman problem with a non-negligible advantage by using $\mathcal{D}$ as a subroutine.

Let (P, aP, bP, cP, z) be a random instance of the problem. $\mathcal{B}$'s goal is to decide whether $z = \hat{e}(P, P)^{abc}$ or not. $\mathcal{B}$ plays the role of $\mathcal{D}$'s challenger in the game of definition 3. At the beginning of this game, $\mathcal{B}$ fixes the system's parameters as in the proof of theorem 1 with $P_{pub} = cP \in \mathbb{G}_1$. These system parameters are given to $\mathcal{D}$ that then performs a polynomially bounded number of queries as explained in definition 3. As in the proof of theorem 1, we assume that any signature query or confirmation/denial query on an identity is preceded by a H_1 oracle query on that identity. We now detail how $\mathcal{B}$ dealt with queries made by $\mathcal{D}$. As in the previous theorem, $\mathcal{B}$ maintains lists L_{H_1}, L_{H_2} and L_{H_3} to keep track of the answers given to hash oracle queries.

- H_1 oracle queries: are treated by $\mathcal{B}$ exactly as in the proof of theorem 1.
- H_2 oracle queries: at any time $\mathcal{D}$ can ask the hash value of a tuple $(M, r, ID) \in \{0,1\}^* \times \{0,1\}^l \times \{0,1\}^*$. When receiving such a query, $\mathcal{B}$ first checks if L_{H_2} contains a tuple (M, r, ID, d, Y) for some $d \in \mathbb{Z}_q$. If it does, $\mathcal{B}$ returns $dP \in \mathbb{G}_1$ as an answer to $\mathcal{D}$ if $Y = 0$ and $d(aP)$ if $Y = 1$. Otherwise, $\mathcal{B}$ picks a random $d \leftarrow_R \mathbb{Z}_q$, inserts the tuple $(M, r, ID, d, 0)$ into L_{H_2} and returns $dP \in \mathbb{G}_1$ as an answer to the query.
- H_3 oracle queries are treated in the simplest way: $\mathcal{B}$ first checks if list L_{H_3} contains an entry indicating that a same query was previously issued. In this case, $\mathcal{B}$ returns the same answer as for the previous query. Otherwise, it answers by a uniformly chosen random element of the appropriate range, namely $\mathbb{Z}_q$.
- key extraction queries are handled by $\mathcal{B}$ exactly as in the proof of theorem 1.
- signature queries are handled by $\mathcal{B}$ exactly as in the proof of theorem 1. As in the latter, $\mathcal{B}$ can always provide a consistent answer to this kind of query.
- confirmation/denial queries: at any time, $\mathcal{D}$ can produce a tuple (M, ID, σ), where $\sigma = (r, \gamma) \in \{0,1\}^l \times \mathbb{G}_2$, and ask for a proof that σ is a valid or invalid signature on M for the signer of identity ID. Unlike what happens in the proof of theorem 1, $\mathcal{B}$ is able to provide $\mathcal{D}$ with a consistent view with overwhelming probability. Most of the time, it can reconstruct the legitimate signature (r, γ') on M for the signer of identity ID with the random string r (this is due to the way that H_2 oracle queries are handled). The confirmation/denial protocol is simulated exactly as in the proof of theorem 1.

After a first series of queries, $\mathcal{D}$ produces a message M an identity ID of its choice on which it wishes to be challenged and requests a challenge for that identity. Recall that it is not allowed to chose an identity for which it asked the corresponding private key at the first stage. $\mathcal{B}$ then builds the challenge like this: it takes a random binary string $r \in \{0,1\}^l$ and checks if an entry $(M, r, ID, .)$ exists in L_{H_2}. If such an entry exists, $\mathcal{B}$ picks another binary string r and repeats the process until finding one such that no entry $(M, r, ID, .)$ is in L_{H_2}. Once an acceptable r is found, $\mathcal{B}$ defines the hash value $H_2(M, r, ID)$ to be $d(aP)$ for a randomly chosen $d \leftarrow_R \mathbb{Z}_q$. The entry $(M, r, ID, d, 1)$ is then inserted into L_{H_2}. Because of the assumptions made above, an entry (ID, μ, X) must exist in L_{H_1} for some $\mu \in \mathbb{Z}_q$. If $X = 0$, then $\mathcal{B}$ stops and outputs "failure". Otherwise, $\mathcal{B}$ computes $\gamma = z^{d\mu} \in \mathbb{G}_2$ and sets the challenge signature as $< r, \gamma >$. It is easy to see that, if $\mathcal{D}$ is a good distinguisher and if z actually equals $\hat{e}(P, P)^{abc}$, then $< r, \gamma >$ must appear as a valid signature for the pair (M, ID) chosen by $\mathcal{D}$ in its challenge request.

At the second stage of the game, $\mathcal{D}$ performs a second series of queries as in the first stage with the restriction that it is now disallowed to ask the private key associated to ID nor to perform a confirmation/denial query for the challenge $(M, ID, < r, \gamma >)$. Because of this fact, $\mathcal{B}$ is able to handle confirmation or denial queries with overwhelming probability: any signature $(M, ID, < r, \gamma' >)$ with $\gamma \neq \gamma'$ is declared to be invalid and the denial protocol is then simulated. The only case where $\mathcal{F}$ is provided with an inconsistent view on a confirmation/denial query is the situation where (P, aP, bP, cP, z) is not a DBDH tuple and where $\mathcal{F}$ queries the confirmation/denial oracle on a tuple $(M, ID, < r, \gamma' >)$ for which $< r, \gamma' >$ is a legitimate signature on M for the identity ID. This event only occurs with negligible probability, since according to theorem 1, we assumed that no algorithm is able to produce a valid signature for a chosen message on a chosen identity with a non-negligible advantage without knowing the private key.

At the end of the game, $\mathcal{D}$ outputs a bit b' that is 0 if it finds that $(M, ID, < r, \gamma >)$ is a valid tuple message-identity-signature and 1 if it finds that $< r, \gamma >$ is a random element of the signature space. If $b' = 0$, then $\mathcal{B}$ outputs 1 as a result to indicate that (P, aP, bP, cP, z) is a valid DBDH tuple. If $b' = 1$, it outputs 0 to indicate that z is a random element of $\mathbb{G}_2$. One can easily verify that, if $\mathcal{D}$ succeeds in distinguishing whether the challenge was a valid or an invalid signature, then $\mathcal{B}$ succeeds in distinguishing DBDH tuples.

It remains to assess $\mathcal{B}$'s probability not to achieve a state of failure. Since a failure of $\mathcal{B}$ can only happen if $\mathcal{B}$ issues a 'bad' key extraction query during the game or if $\mathcal{D}$ chooses to be challenged on a 'bad' identity, the probability for $\mathcal{B}$ not to fail is at least $\delta_1^{q_E}(1-\delta)$ which is at least $1/e(q_e+1)$ if the optimal value of δ_1 is used by $\mathcal{B}$ in its strategy to handle H_1 queries. From this, it comes that if ϵ denotes $\mathcal{D}$'s advantage as a distinguisher, then $\mathcal{B}$'s advantage in distinguishing DBDH tuples is at least $\epsilon/e(q_E + 1)$.

$\square$