New identity based signcryption schemes from pairings

Benoît Libert Jean-Jacques Quisquater

UCL, Microelectronics Laboratory, Crypto Group
Place du Levant, 3, B-1348, Louvain-La-Neuve, Belgium
Telephone: +32(0)10 47.80.62, Fax: +32(0)10 47.25.98
{libert,quisquater}@dice.ucl.ac.be

Abstract. We present a new identity based scheme based on pairings over elliptic curves. It combines the functionalities of signature and encryption and is provably secure in the random oracle model. We compare it with Malone-Lee's one from security and efficiency points of view. We give a formal proof of semantical security under the Decisional Bilinear Diffie-Hellman assumption for this new scheme and we show how to devise other provably secure schemes that produce even shorter ciphertexts.

1 Introduction

Identity based cryptosystems were introduced by Shamir in 1984 ([24]). The idea was to get rid of public key certificates by allowing the user's public key to be the binary sequence corresponding to an information identifying him in a non ambiguous way (e-mail address, IP address combined to a user name, social security number,...). This kind of system allows to avoid trust problems encountered in certificate based public key infrastructures (PKIs): there is no need to bind a public key to its owner's identity since those are one single thing. These systems have the particularity to involve trusted authorities called private key generators (PKGs) whose task is to compute users'private key from their identity information (users do not generate their key pairs themselves). Several practical identity based signature schemes (IBS) have been devised since 1984 ([9], [12]) but a satisfying identity based encryption scheme (IBE) only appeared in 2001 ([5]). It was devised by Boneh and Franklin and cleverly uses bilinear maps (the Weil or Tate pairing) over supersingular elliptic curves. Other identity based schemes using pairings were proposed after 2001 ([6],[25],[14],[29],[8]).

The concept of public key signcryption schemes was found by Zheng in 1997 ([31]). The idea of this kind of primitive is to perform encryption and signature in a single logical step in order to obtain confidentiality, integrity, authentication and non-repudiation more efficiently than the sign-then-encrypt approach. The drawback of this latter solution is to expand the final ciphertext size (this could be impractical for low bandwidth networks) and increase the sender and receiver's computing time. Several efficient signcryption schemes have been proposed since 1997 ([30],[32],[33],[28],[27]) and a first example of formal security proof in a formal security model was published in 2002 ([2]). However, until 2002, none of these schemes were identity based. B. Lynn extended the Boneh-Franklin scheme to build an efficient authenticated IBE scheme ([17]) but his scheme does not have the non-repudiation property. The recipient of a ciphertext is the only entity to be able to check the ciphertext's origin and validity. J. Malone-Lee recently proposed a first method to achieve an identity based signcryption solution ([18]). In this paper, we analyze Malone-Lee's system and compare it with a new scheme that can be somewhat more efficient and produces shorter ciphertexts.

1.1 General scheme

Just like classical IBE systems, signcryption schemes are made of four algorithms which are the following.

Setup : given a security parameter k, the private key generator (PKG) generates the system's public parameters *params*.
Keygen : given an identity ID, the PKG computes the corresponding private key d_{ID} and transmits it to its owner in a secure way.
Signcrypt : to send a message m to Bob, Alice computes Signcrypt(m, d_{ID_a}, ID_b) to obtain the ciphertext σ.
Unsigncrypt : when Bob receives σ, he computes Unsigncrypt(σ, d_{ID_b}, ID_a) and obtains the clear text m or the symbol $\perp$ if σ was an invalid ciphertext between identities ID_a and ID_b.

For consistency purposes, we of course require that if $\sigma =$ Signcrypt(m, d_{ID_a}, ID_b), then we have $m =$ Unsigncrypt(σ, d_{ID_b}, ID_a).

1.2 Security notions

Malone-Lee defines extended security notions for identity based signcryption schemes (IBSC). These notions are semantical security (i.e. indistinguishability against adaptive chosen cipher text attacks) and unforgeability against adaptive chosen messages attacks. Formally, we recall the following definitions.

Definition 1. *We say that an identity based signcryption scheme (IDSC) has the indistinguishability against adaptive chosen ciphertext attacks property (IND-IDSC-CCA) if no polynomially bounded adversary has a non-negligeable advantage in the following game.*

1. *The challenger runs the Setup algorithm with a security parameter k and sends the system parameters to the adversary.*
2. *The adversary $\mathcal{A}$ performs a polynomially bounded number of requests:*
   - *Signcryption request: $\mathcal{A}$ produces two identities ID_i, ID_j and a plaintext M. The challenger computes $d_{ID_i} = Keygen(ID_i)$ and then $Signcrypt(m, d_{ID_i}, ID_j)$ and sends the result to $\mathcal{A}$.*
   - *Unsigncryption request: $\mathcal{A}$ produces two identities ID_i and ID_j, a ciphertext σ. The challenger generates the private key $d_{ID_i} = Keygen(ID_i)$ and sends the result of $Unsigncrypt(\sigma, d_{ID_j}, ID_i)$ to $\mathcal{A}$ (this result can be the $\perp$ symbol if σ is an invalid ciphertext).*
   - *Key extraction request: $\mathcal{A}$ produces an identity ID and receives the extracted private key $d_{ID} = Keygen(ID)$.*

   $\mathcal{A}$ can present its requests adaptively: every request may depend on the answer to the previous ones.
3. *$\mathcal{A}$ chooses two plaintexts $M_0, M_1 \in \mathcal{M}$ and two identities ID_A and ID_B on which he wishes to be challenged. He cannot have asked the private key corresponding to ID_A nor ID_B in the first stage.*
4. *The challenger takes a bit $b \in_R \{0,1\}$ and computes $C = Signcrypt(M_b, d_{ID_A}, ID_B)$ which is sent to $\mathcal{A}$.*

5. *$\mathcal{A}$ asks again a polynomially bounded number of requests just like in the first stage. This time, he cannot make a key extraction request on ID_A nor ID_B and he cannot ask the plaintext corresponding to C.*
6. *Finally, $\mathcal{A}$ produces a bit b' and wins the game if $b' = b$.*

The adversary's advantage is defined to be $Adv(\mathcal{A}) := \big|2P[b' = b] - 1\big|$.

Definition 2. *An identity based signcryption scheme (IDSC) is said to be secure against an existential forgery for adaptive chosen messages attacks (EF-IDSC-ACMA) if no polynomially bounded adversary has a non-negligeable advantage in the following game.*

1. *The challenger runs the Setup algorithm with a security parameter k and gives the system parameters to the adversary.*
2. *The adversary $\mathcal{A}$ performs a polynomially bounded number of requests just like in the previous definition.*
3. *Finally, $\mathcal{A}$ produces a new triple (σ^*, ID_A, ID_B) (i.e. a triple that was not produced by the signcryption oracle), where the private key of ID_A was not asked in the second stage and wins the game if the result of Unsigncrypt$(\sigma^*, d_{ID_A}, ID_B)$ is not the $\perp$ symbol.*

The adversary's advantage is simply its probability of victory.

In this definition, the adversary is allowed to ask the private key corresponding to the identity ID_B for which the ciphertext he produces must be valid. This condition is necessary to obtain the non-repudiation property and to prevent a dishonest recipient to send a ciphertext to himself on Alice's behalf and to try to convince a third party that Alice was the sender.

1.3 Preliminaries

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

1. Bilinearity: $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{F}_q^*, \quad \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
2. Non-degeneracy: for any point $P \in \mathbb{G}_1$, $\hat{e}(P, Q) = 1$ for all $Q \in \mathbb{G}_1$ iff $P = \mathcal{O}$.
3. Computability: there exists an efficient algorithm to compute $\hat{e}(P, Q)\ \ \forall\ P, Q \in \mathbb{G}_1$.

The modified Weil pairing and the Tate pairing are admissible applications. $\mathbb{G}_1$ is a cyclic 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 security of the schemes described here relies on the hardness of the following problems.

Definition 3. *Given two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$, the **Bilinear Diffie-Hellman problem** (BDHP) 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** (DBDHP) is, given a tuple of points (P, aP, bP, cP) and an element $h \in \mathbb{G}_2$, to decide whether $h = \hat{e}(P, P)^{abc}$ or not.*

We define the advantage of a distinguisher against the DBDH problem like this

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

The decisional problem is of course not harder than the computational one. However, no algorithm is known to be able to solve any of them so far.

2 The Malone-Lee signcryption scheme

Setup : given a security parameter k, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, 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 $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \mathbb{F}_q^*$, $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^n$. It chooses a master-key $s \in \mathbb{F}_q^*$ and computes $P_{pub} = sP$. The system's public parameters are

$$\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, n, \hat{e}, P, P_{pub}, H_1, H_2, H_3)$$

where n denotes the size of plaintexts.

Keygen : given an identity ID, the PKG computes $Q_{ID} = H_1(ID)$ and the private key $d_{ID} = sQ_{ID}$.

Signcrypt : to send a message M to Bob, Alice follows the steps below

1. Compute $Q_{ID_b} = H_1(ID_b) \in \mathbb{G}_1$,
2. Choose $x \leftarrow_R \mathbb{F}_q^*$, and compute $U = xP$, $r = H_2(U||m) \in \mathbb{F}_q^*$.
3. Compute $\hat{e}(P_{pub}, Q_{ID_b})^x \in \mathbb{G}_2$ and $V = xP_{pub} + rd_{ID_a} \in \mathbb{G}_1$.
4. Compute $\kappa = H_3(\hat{e}(P_{pub}, Q_{ID_b})^x)$.
5. Computes $c = m \oplus \kappa$.

The ciphertext is $C = (U, V, c)$.

Unsigncrypt : when receiving $C = (U, V, c)$, Bob performs the following tasks

1. Compute $Q_{ID_a} = H_1(ID_a) \in \mathbb{G}_1$
2. Compute $y = \hat{e}(U, d_{ID_b})$ and $\kappa = H_3(y)$
3. Recover $m = \kappa \oplus c$.
4. Compute $r = H_2(U||m)$ and accept C if and only if

$$\hat{e}(V, P) = \hat{e}(Q_{ID_a}, P_{pub})^r \hat{e}(U, P_{pub}).$$

The consistency is easy to verify since the κ computed by Bob is the same as Alice's one:

$$\hat{e}(U, d_{ID_b}) = \hat{e}(xP, sQ_{ID_b}) = \hat{e}(xP_{pub}, Q_{ID_b}).$$

Once Bob has recovered m, he can prove to a third party that the ciphertext $\sigma = (c, U, V)$ was created by Alice. The third party simply computes $r = H_2(U||m)$. If the ciphertext actually comes from Alice, we have

$$\hat{e}(V, P) = \hat{e}(rd_{ID_a} + xP_{pub}, P) = \hat{e}(d_{ID_a}, P)^r \hat{e}(xsP, P) = \hat{e}(Q_{ID_a}, P)^r \hat{e}(U, P_{pub})$$

and the condition of the 4th step of Unsigncrypt is verified. Given the system's public parameters, a plaintext m and the corresponding ciphertext σ, anybody can verify the origin of σ.

We notice that the scheme is the result of a combination of the simplified version of Boneh and Franklin's IBE cryptosystem (see [5] for details: this is the version that only has the ID-OWE level of security) with the following signature :

Setup and **Keygen** are the same as above.

Sign : to sign a message M,

1. Choose $x \leftarrow_R \mathbb{F}_q^*$ and compute $U = xP$.
2. Compute $r = H(m, U)$
3. Compute $V = xP_{pub} + rd_{ID_A}$

The signature on M is $\sigma = (U, V)$

Verify :

1. Compute $r = H(M, U)$
2. Accept the signature iff

$$\hat{e}(P, V) = \hat{e}(U, P_{pub})\hat{e}(P_{pub}, Q_{ID_A})^r$$

This signature may be viewed as a variant of Hess's identity based signature ([14]). The ciphertexts produced by Malone-Lee's scheme are nearly a concatenation of a signature and a ciphertext (this approach is often called "encrypt-and-sign" in the literature) and only spares one point multiplication in $\mathbb{G}_1$ compared to the encrypt-and-sign approach. Furthermore, this scheme cannot achieve the semantical security. Indeed, as pointed out in [26], as soon as the signature on the plaintext is visible in the ciphertext, the scheme cannot be semantically secure because any attacker can simply verify the signature on plaintexts m_0 and m_1 produced during the game IND-IDSC-CCA and find out which one matches to the challenge ciphertext. Although, it can offer a reasonable security, the scheme is not semantically secure in its current version. All systems resulting from the encrypt-and-sign approach have the same inherent weakness. Now, it becomes clear that the universal verifiability feature of signcryption schemes can hamper their resistance to chosen-ciphertext attacks. Shin, Lee and Shin described in [26] how to solve this problem using a modified DSA-type signature. Malone-Lee and Mao also proposed in [19] a secure verifiable signcryption scheme based on RSA but both of these two schemes cannot support identity based public keys. We describe in the next section an identity based signcryption scheme that achieves both public verifiability and resistance to chosen-ciphertext attacks.

3 A new identity based signcryption scheme

In [33], Zheng showed how to use the SDSS1 and SDSS2 [1] signatures schemes to build efficient signcryption schemes. He pointed out that his construction can use any shortened El Gamal based signature scheme or the Schnorr signature scheme (without any shortening) to provide a signcryption solution. Unfortunately, his scheme does not offer the non-repudiation property and we have to use a modification of a construction due to Bao and Deng ([3]) to obtain it. We show that a variant of Hess's identity based signature [2] can also be used as a building block to obtain efficient provably secure identity based signcryption schemes. We will give a proof of semantical security under a reasonable computational assumption which is the hardness of the DBDH problem.

[1] Shortened Digital Signature Standard: these schemes were obtained by applying to DSS a method for shortening El Gamal based signatures

[2] The identity based signature scheme proposed by Hess at SAC 2002 is an adaptation of Schnorr's signature that uses the Tate pairing rather than exponentiation as a group isomorphism. It is shown to be secure against existential forgery under adaptive chosen-messages attacks in the random oracle model.

3.1 Description of the scheme

Setup : given security parameters k and n, the PKG chooses the same system parameters as in the Malone-Lee scheme except that it also chooses a secure symmetric cipher (E, D) and hash functions are now $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$, and $H_3 : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{F}_q$.	**Keygen :** given an identity ID, the PKG computes $Q_{ID} = H_1(ID)$ and the private key $d_{ID} = sQ_{ID}$
Signcrypt : to send a message m to Bob, Alice follows the steps below 1. Compute $Q_{ID_B} = H_1(ID_B) \in \mathbb{G}_1$. 2. Choose $x \leftarrow_R \mathbb{F}_q^*$, and compute $k_1 = \hat{e}(P, P_{pub})^x$ and $k_2 = H_2(\hat{e}(P_{pub}, Q_{ID_B})^x)$. 3. Compute $c = E_{k_2}(m)$, $r = H_3(c, k_1)$ $S = xP_{pub} - rd_{ID_A} \in \mathbb{G}_1$. The ciphertext is $\sigma = (c, r, S)$.	**Unsigncrypt :** when receiving $\sigma = (c, r, S)$, Bob performs the following tasks 1. Compute $Q_{ID_A} = H_1(ID_A) \in \mathbb{G}_1$ 2. Compute $k_1 = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r$ 3. Compute $\tau = \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^r$ and $k_2 = H_2(\tau)$. 4. Recover $m = D_{k_2}(c)$ and accept σ if and only if $r = H_3(c, k_1)$.

The consistency is easy to verify by the bilinearity of the map. We have

$$\hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r = \hat{e}(P, P_{pub})^x \quad \text{and} \quad \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^r = \hat{e}(P_{pub}, Q_{ID_B})^x.$$

Any third party (like firewalls as explained in [11]) can be convinced of the message's origin by recovering k_1 just like in step 1 above and checking if the condition $r = H(c, k_1)$ holds. The knowledge of the plaintext m is not required for the public verification of a message's origin. In order to convince someone that Alice is the sender of a particular plaintext m, the receiver just has to forward the ephemeral decryption key k_2 to the third party.

It is important to notice that replacing $r = H_3(c, k_1)$ by $r = H_3(m, k_1)$ would induce the same obstacle to the semantical security as in the Malone-Lee scheme (see [26]). It is shown in [11] that this modification of the Bao-Deng original construction does not affect the unforgeability of the system and one can show that it can enhance the security of the original Bao-Deng scheme to the semantical level.

This scheme is as efficient as Malone-Lee's method (since the pairing $\hat{e}(P, P_{pub})$ does not depend on users or messages and can always be precomputed) and it can be slightly more efficient when users often have to communicate between each other (pairings $\hat{e}(P_{pub}, Q_{ID_B})$ and $\hat{e}(Q_{ID_A}, d_{ID_B})$ can be precomputed by the sender and the receiver once for all). In this case, the most expensive operations of the signcryption algorithm are two exponentiations in $\mathbb{G}_2$ and one computation of the type $aP + bQ \in \mathbb{G}_1$. The Unsigncrypt operation only requires two pairing evaluations and two exponentiations. When precomputing is done with the Malone-Lee scheme, we have a similar cost for the signcryption but we need three pairing evaluations for the unsigncryption. Furthermore, we will show that the above scheme does not need the Fujisaki-Okamoto transformation ([10]) to achieve the semantical security (under a stronger assumption nevertheless).

3.2 Security

Theorem 1. *In the random oracle model, we assume we have an IND-IDSC-CCA adversary called $\mathcal{A}$ that is able to distinguish ciphertexts during the game of definition 1 with an advantage ϵ when running in a time t and asking at most q_{H_1} identity hashing requests, at most q_R H_3 requests, q_R Signcrypt requests and q_U Unsigncrypt requests. Then, there exists a distinguisher $\mathcal{B}$ that can solve the Decisional Bilinear Diffie-Hellman problem in a time $O(t + (8q_R^2 + 4q_U)T_{\mathcal{E}})$ with an advantage*

$$Adv(\mathcal{B})^{DBDH(\mathbb{G}_1,P)} > 2(\epsilon - q_U/2^{k-1})/q_{H_1}^4$$

where $T_{\mathcal{E}}$ denotes the computation time of the bilinear map.

Proof. see the appendix.

□

It is possible to prove the semantical security under the weaker Bilinear Diffie-Hellman assumption by applying to the scheme the Fujisaki-Okamoto transformation but as far as the decisional problem is believed to be hard, we think it is better to use the system in its original form.

The unforgeability against adaptive chosen messages attacks derives from the security of Hess's identity based signature scheme ([14]) under the computational Diffie-Hellman assumption. By arguments similar to those in [11], one can show that an attacker that is able to forge a signcrypted message must be able to forge a signature for the following scheme which is a variant of Hess's signature.

Setup and **Keygen** are the same as above.

Sign : to sign a message M,

1. Choose $x \leftarrow_R \mathbb{F}_q^*$ and compute $k = \hat{e}(P, P_{pub})^x$.
2. Compute $r = H(m, k)$
3. Compute $S = xP_{pub} - rd_{ID_A}$

The signature on M is $\sigma = (r, S)$

Verify : when receiving M and σ,

1. Compute $k' = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r$
2. Accept the signature iff $r = H(m, k')$

4 Further results

It is possible to shorten the ciphertext produced by our scheme described in section 3 by slightly modifying it using a construction from [16].

Setup and **Keygen :** remain unchanged except that we do not use any symmetric cipher and hash functions are now

$$H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1, \quad H_2 : \mathbb{G}_2 \rightarrow \mathbb{F}_q, \quad H_3 : \{0,1\}^n \times \mathbb{F}_q \rightarrow \{0,1\}^m.$$

The plaintexts must have a fixed bitlength of n with $m + n < k \approx \log_2 q$.

Signcrypt : to send a message M to Bob, Alice follows the steps below

1. Compute $Q_{ID_B} = H_1(ID_B) \in \mathbb{G}_1$.
2. Choose $x \leftarrow_R \mathbb{F}_q^*$, and compute $k_1 = H_2(\hat{e}(P, P_{pub})^x)$ and $k_2 = H_2(\hat{e}(P_{pub}, Q_{ID_B})^x)$.
3. Compute $r{=}(M||H_3(M, k_2))k_1k_2 \mod q$, and $S = xP_{pub} - rd_{ID_A} \in \mathbb{G}_1$.

The ciphertext is $\sigma = (r, S)$.

Unsigncrypt : when receiving $\sigma = (r, S)$, Bob performs the following tasks

1. Compute $Q_{ID_A} = H_1(ID_A) \in \mathbb{G}_1$.
2. Compute $k_1 = H_2(\hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r)$.
3. Compute $\tau = \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^r$ and $k_2 = H_2(\tau)$.
4. Compute $M' = r(k_1k_2)^{-1} \mod q$.
5. Take M_1 as the first n bits of M' and accept the plaintext M_1 if and only if $(M_1, H_3(M_1, k_2))$ are the first $m + n$ bits of M'.

To convince a third party that Alice created the ciphertext, Bob simply forwards it k_2, r, and S. The third party just has to compute k_1 as in step 2 of Unsigncrypt and check if $M' = r(k_1k_2)^{-1} \mod q$ satisfies the condition of step 5.

As explained in [16], it is necessary to include k_2 in the hashing during the computation of r to prevent Bob to use a received ciphertext for creating a new one on Alice's behalf. The security of the scheme can be proven under the DBDH assumption while the unforgeability can be proven in the same way as in [17] under the BDH assumption.

The present scheme avoids the use of symmetric ciphers but requires some additional arithmetic operations in $\mathbb{F}_q$. Its disadvantage is indeed the short size of plaintexts since we must have $(M||H_3(M, k_2)) \in \mathbb{F}_q$. That is the price to pay to obtain short ciphertexts. A solution described in [16] is to split the plaintext into blocks of n bits before encrypting it but it increases the communication costs. However, the system remains practical for short messages like symmetric keys and can be used as a key transport mechanism.

The security of the latter scheme also relies on the intractability of the DBDH problem. We can state a theorem similar to theorem 1.

Theorem 2. *In the random oracle model, we assume we have an IND-IDSC-CCA adversary called $\mathcal{A}$ that is able to distinguish ciphertexts during the game of definition 1 with an advantage ϵ when running in a time t and asking at most q_{H_1} identity hashing requests, at most q_{H_3} H_3 queries and signcryption queries and q_U unsigncryption queries. Then, there exists a distinguisher $\mathcal{B}$ that can solve the Decisional Bilinear Diffie-Hellman problem in a time $O(t + (8q_{H_3}^2 + 4q_U)T_{\mathcal{E}})$ with an advantage*

$$Adv(\mathcal{B})^{DBDH(\mathbb{G}_1,P)} > 2(\epsilon - q_U/2^{k-1})/q_{H_1}^4$$

where $T_{\mathcal{E}}$ denotes the computation time of the bilinear map.

Proof. see the appendix.

□

We obtain the same bound on the probability of success as in theorem 1 but the simulations of the reduction are a little different.

Finally, the DBDH assumption allows to simplify an existing scheme. If we consider Lynn's authenticated identity based encryption scheme ([17]), we state it can be simplified because the involved pairing $\hat{e}(d_{ID_A}, Q_{ID_B}) = \hat{e}(Q_{ID_A}, d_{ID_B})$ allows to build a security proof based on the DBDH problem (rather than the BDH one) similar to the proof of the previous section. One further obtains a better reduction and this simplification results in a quite efficient authenticated IBE scheme that produces shorter ciphertexts than Lynn's original one.

Setup and **Keygen** do not change.

Authenticrypt : to send a message M to Bob, Alice follows the steps below	**Authentidecrypt :** when receiving $\sigma = (U, V)$, Bob performs these tasks
1. Compute $Q_{ID_B} = H_1(ID_B) \in \mathbb{G}_1$,	1. Compute $Q_{ID_A} = H_1(ID_A) \in \mathbb{G}_1$ and $g_{AB} = \hat{e}(Q_{ID_A}, d_{ID_B})$.
2. Choose $r \leftarrow_R \mathbb{F}_q$	2. Recover $M = V \oplus H_2(U, g_{AB})$
3. Compute $g_{AB} = \hat{e}(d_{ID_A}, Q_{ID_B})$.	3. Reject the plaintext if it does not contain the appropriate redundancies.
The ciphertext is $C = (r, M \oplus H_2(r, g_{AB}))$.	

The unforgeability of this simplified scheme still relies on the BDH assumption. The proof in [17] is not modified. As its original form, this scheme does not prevent a receiver to send messages to himself on behalf to the sender but it does not matter since there is no public verifiability: the receiver can never impersonate the sender from a third party's point of view.

The schemes described so far in this paper do not provide any forward secrecy functionality. If the sender or the receiver's private key are compromised, the attacker can recover each of the issued messages by using the equality $\hat{e}(Q_{ID_A}, d_{ID_B}) = \hat{e}(d_{ID_A}, Q_{ID_B})$. As done in [15] with Zheng's scheme, it is possible to modify our system of section 3.1 to obtain some forward security features at the price of losing the universal verifiability. The result is an authenticated encryption scheme with forward secrecy. The only modifications consist in disallowing public verifiability and replacing r by rQ_{ID_A} in the ciphertext. Formally, the Setup and Keygen algorithms do not change while the other ones look like below.

Authenticrypt : to send a message m to Bob, Alice follows the steps below	**Authentidecrypt :** when receiving $\sigma = (c, R, S)$, Bob performs these tasks
1. Compute $Q_{ID_B} = H_1(ID_B) \in \mathbb{G}_1$.	1. Compute $Q_{ID_A} = H_1(ID_A) \in \mathbb{G}_1$.
2. Choose $x \leftarrow_R \mathbb{F}_q^*$, and compute $(k_1, k_2) = H_2(\hat{e}(P_{pub}, Q_{ID_B})^x)$.	2. Compute $\tau = \hat{e}(S, Q_{ID_B})\hat{e}(R, d_{ID_B})$ and $(k_1, k_2) = H_2(\tau)$.
3. Compute $c = E_{k_2}(m)$, $r = H_3(c, k_1)$, $S = xP_{pub} - rd_{ID_A} \in \mathbb{G}_1$ and $R = rQ_{ID_A}$.	3. Recover $m = D_{k_2}(c)$.
	4. Compute $r = H_3(c, k_1)$ and rQ_{ID_A}.
	5. Accept σ if and only if $R = rQ_{ID_A}$.
The ciphertext is $\sigma = (c, R, S)$.	

Even when knowing d_{ID_A}, an attacker is still unable to find out the plaintext since he does not know r. The proof of semantical security security described in the appendix can easily be adapted to prove the resistance of the modified scheme to adaptive chosen-ciphertext attacks.

So far, it remains an open problem to devise an efficient signcryption scheme providing both public verifiability and forward security.

5 Conclusions

We have shown that Hess's signature together with a modification of the Bao-Deng construction can be used to build a new efficient identity based signcryption scheme that provides a better security than Malone-Lee's scheme. Both systems are more efficient than the approach consisting in combining the Boneh-Franklin encryption scheme with a signature. Our solution satisfies the semantic security notion under the Decisional Bilinear Diffie-Hellman assumption. Although, this is a stronger assumption than the difficulty of the computational bilinear problem, it seems to be a reasonable base for the security of cryptosystems. We showed it allows to build other efficient schemes and it probably has other applications.

It might be possible to build secure identity based signcryption schemes that are even more efficient than ours but a possible goal for future research would be to find hierarchical ID-based signcryption schemes that allow users of a system to receive signcrypted messages from senders who do not depend on the same authority.

Another interesting open question is the possible equivalence between the Decisional Bilinear Diffie-Hellman problem and the computational one.

Acknowledgements

We would like to thank Jun-Bum Shin for helpful discussions.

References

1. J.H. An, Y. Dodis, T. Rabin, *On the security of joint signature and encryption*, Advances in Cryptology - Eurocrypt'02, LNCS 2332, Springer, pp. 83-107, 2002.
2. J. Baek, R. Steinfeld, Y. Zheng, *Formal Proofs for the Security of Signcryption*, Proc. of PKC'02, LNCS 2274, Springer, pp. 81-98.
3. F. Bao, R.H. Deng, *A signcryption scheme with signature directly verifiable by public key*, Proc. of PKC'98, LNCS 1431, Springer, pp. 55-59, 1998.
4. 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 Secrurity, pp. 62-73, 1993.
5. D. Boneh, M. Franklin, *Identity Based Encryption From the Weil Pairing*, Advances in Cryptology - Crypto'01, LNCS 2139, Springer, 2001.
6. J.C. Cha, J.H. Cheon, *An Identity-Based Signature from Gap Diffie-Hellman Groups*, to appear in proceedings of PKC 2003. Springer Verlag, Lecture Notes in Computer Science series.
7. C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Proc. of Cryptography and Coding, LNCS 2260, Springer, pp. 360-363, 2001.
8. R. Dupont, A. Enge, *Practical Non-Interactive Key Distribution Based on Pairings*, available at http://eprint.iacr.org/2002/136.
9. A. Fiat, A. Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems* , Advances in Cryptology - Crypto'86, LNCS 0263, Springer, pp. 186-194, 1986.
10. E. Fujisaki, T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes", Advances in Cryptology - Crypto'99, LNCS 1666, Springer, pp.537-554, 1999.
11. C. Gamage, J. Leiwo, Y. Zheng, *Encrypted message authentication by firewalls*, Proc. of PKC'99, LNCS 1560, Springer, pp. 69-81, 1999.
12. L. Guillou, J-J. Quisquater, *A "Paradoxical" Identity-Based Signature Scheme Resulting From Zero-Knowledge* , Advances in Cryptology - Crypto'88, LNCS 0403, Springer, pp. 216-231, 1988
13. W.H. He, T.C. Wu, *Cryptanalysis and improvement of Petersen-Michels signcryption scheme*, IEE Proc. - Computers and Digital Techniques, 146(2), pp. 123-124, March 1999.
14. F. Hess, *Efficient identity based signature schemes based on pairings*, to appear in proceedings of SAC 2002. Springer Verlag, Lecture Notes in Computer Science series.
15. H.Y. Jung, D.H. Lee, J.I. Lim, K.S. Chang, *Signcryption Schemes with Forward Secrecy*, Proc. of WISA'01, available at http://cist.korea.ac.kr/Tr/TR01_6.pdf
16. M.K. Lee, D.K. Kim, K. Park, *An Authenticated Encryption Scheme with Public Verifiability*, in 4^{th} Korea-Japan Joint Workshop on Algorithms and Computation, 2000.
17. B. Lynn, *Authenticated Identity-Based Encryption*, available at http://eprint.iacr.org/2002/072/.

18. J. Malone-Lee, *Identity Based Signcryption*, available at http://eprint.iacr.org/2002/098/.
19. J. Malone-Lee, W. Mao, *Two Birds one Stone: Signcryption using RSA*, to appear in proceedings of CT-RSA 2003. Springer Verlag, Lecture Notes in Computer Science series.
20. K.G. Paterson, *ID-based signatures from pairings on elliptic curves*, available at http://eprint.iacr.org/2002/004/.
21. D. Pointcheval, J. Stern, *Security proofs for signature schemes*, Advances in Cryptology - Eurocrypt'96, LNCS 1070, Springer, pp. 387-398, 1996.
22. D. Pointcheval, J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptology, vol. 13-Number 3, pp. 361-396, 2000.
23. R. Sakai, K. Ohgishi, M. Kasahara, *Cryptosystems based on pairing*, In The 2000 Sympoium on Cryptography and Information Security, Okinawa, Japan, January 2000.
24. A. Shamir, *Identity Based Cryptosystems and Signature Schemes*, Advances in Cryptology - Crypto' 84, LNCS 0196, Springer, 1984.
25. N.P. Smart, *An identity based authenticated key agreement protocol based on the Weil pairing*, Electronic Letters, 38(13): 630-632, 2002.
26. J-B. Shin, K. Lee, K. Shim, *New DSA-verifiable signcryption schemes*, to appear in proceedings of ICISC 2002. Springer Verlag, Lecture Notes in Computer Science series.
27. R. Steinfeld, Y. Zheng, *A Signcryption Scheme Based on Integer Factorization*, Proc. of ISW'00, pp. 308-322, 2000.
28. B.H. Yum, P.J. Lee, *New Signcryption Schemes Based on KCDSA*, Proc. of ICISC'01, LNCS 2288, Springer, pp. 305-317, 2001.
29. F. Zhang, S. Liu, K. Kim, *ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings*, available at http://eprint.iacr.org/2002/122.
30. Y. Zheng, H. Imai, *Efficient Signcryption Schemes On Elliptic Curves*, Proc. of IFIP/SEC'98, Chapman & Hall, 1998.
31. Y. Zheng, *Digital Signcryption or How to Achieve Cost (Signature & Encryption) $\ll$ Cost (Signature) + Cost (Encryption)*, Advances in Cryptology - Crypto'97, LNCS 1294, Springer, pp. 165-179, 1997.
32. Y. Zheng, *Identification, Signature and Signcryption using High Order Residues Modulo an RSA Composite*, Proc. of PKC'01, LNCS 1992, Springer, pp. 48-63, 2001.
33. Y. Zheng, *Signcryption and its applications in efficient public key solutions*, Proc. of ISW'97, pp. 291-312, 1998.

Appendix

Proof of theorem 1

The distinguisher $\mathcal{B}$ receives a random instance (P, aP, bP, cP, h) of the Decisional Bilinear Diffie-Hellman problem. His goal is to decide whether $h = \hat{e}(P, P)^{abc}$ or not. $\mathcal{B}$ will run $\mathcal{A}$ as a subroutine and act as $\mathcal{A}$'s challenger in the IND-IDSC-CCA game. $\mathcal{B}$ needs to maintain lists L_1, L_2 et L_3 that are initially empty and are used to keep track of answers to queries asked by $\mathcal{A}$ to oracles H_1, H_2 and H_3. The list L_3 will be used to simulate the signcryption oracle. We assume that any Signcrypt or Unsigncrypt request on a pair of identities happens after $\mathcal{A}$ asked the hashing H_1 of these identities. Any key extraction query on an identity is also preceded by a hash query on the same identity. We also assume $\mathcal{A}$ never makes an unsigncryption query on a ciphertext obtained from the signcryption oracle. He only makes unsigncryption queries for observed ciphertexts.

At the beginning of the game, $\mathcal{B}$ gives $\mathcal{A}$ the system parameters with $P_{pub} = cP$ (c is unknown to $\mathcal{B}$ and plays the role of the PKG's master-key). Then, $\mathcal{B}$ chooses two distinct random numbers $i, j \in \{1, \ldots, q_{H_1}\}$. $\mathcal{A}$ asks a polynomially bounded number of H_1 requests on identities of his choice. At the i^{th} H_1 request, $\mathcal{B}$ answers by $H_1(ID_i) = aP$. At the j^{th}, he answers by $H_1(ID_j) = bP$. Since aP and bP belong to a random instance of the DBDH problem, $\mathcal{A}$'s view will not be modified by these changes. Hence, the private keys d_{ID_i} and d_{ID_j} (which are not computable by $\mathcal{B}$) are respectively acP and bcP. Thus the solution $\hat{e}(P, P)^{abc}$ of the BDH problem is given by $\hat{e}(Q_{ID_i}, d_{ID_j}) = \hat{e}(d_{ID_i}, Q_{ID_j})$. For requests $H_1(ID_e)$ with $e \neq i, j$, $\mathcal{B}$ chooses $b_e \leftarrow_R \mathbb{F}_q^*$, puts the pair (ID_e, b_e) in list L_1 and answers $H_1(ID_e) = b_e P$.

We now explain how the other kinds of requests are treated by $\mathcal{B}$.

$\mathbf{H}_2$ requests : on a $H_2(g_e)$ request, $\mathcal{B}$ searches a pair (g_e, R_e) in the list L_2. If such a pair is found, $\mathcal{B}$ answers by R_e, otherwise he answers $\mathcal{A}$ by a random binary sequence $R_e \leftarrow_R \{0,1\}^*$ such that no entry $(., R)$ exists in L_2 (in order to avoid collisions on H_2) and puts the pair (g_e, R) into L_2.

$\mathbf{H}_3$ requests : for a query $H_3(c_e, k_e)$, $\mathcal{B}$ first ensures the list L_3 does not contain a tuple (c_e, k_e, r_e). If such a tuple is found, $\mathcal{B}$ answers r_e, otherwise he chooses $r \leftarrow_R \mathbb{F}_q$, gives it as an answer to the query and puts the tuple (c_e, k_e, r) into L_3.

Key extraction requests : when $\mathcal{A}$ asks a question Keygen(ID_A), if $ID_A = ID_i$ or $ID_A = ID_j$, then $\mathcal{B}$ fails and stops. If $ID_A \neq ID_i, ID_j$ then the list L_1 must contain a pair (ID_A, d) for some d (this indicates $\mathcal{B}$ previously answered $H_1(ID_A) = dP$ on a H_1 query on ID_A). The private key corresponding to ID_A is then $dP_{pub} = cdP$. It is computed by $\mathcal{B}$ and returned to $\mathcal{A}$.

Signcrypt requests : at any time $\mathcal{A}$ can perform a Signcrypt request for a plaintext M and identities ID_A and ID_B. Let us first see what happens if ID_A and ID_B are not the identities ID_i and ID_j. In the case $ID_A \neq ID_i, ID_j$, $\mathcal{B}$ computes the private key d_{ID_A} corresponding to ID_A by running the key extraction request algorithm and then can simply run the algorithm Signcrypt(M, d_{ID_A}, Q_{ID_B}).

In the case $ID_A = ID_i$ or $ID_A = ID_j$ and $ID_B \neq ID_i, ID_j$, $\mathcal{B}$ has to simulate the execution of Signcrypt(M, d_{ID_A}, Q_{ID_B}) as follows. He chooses $r \leftarrow_R \mathbb{F}_q$ and $S \leftarrow_R \mathbb{G}_1^*$. He computes $k' = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r$ and $\tau = \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^r$ where d_{ID_B} is the private key corresponding to ID_B ($\mathcal{B}$ could obtain it from the key extraction algorithm because $ID_B \neq ID_i, ID_j$). He runs the H_2 simulation algorithm to find $k_2 = H_2(\tau)$ and computes $c = E_{k_2}(M)$. He then checks if L_3 already contains a tuple (c, k', r') with $r' \neq r$. In this case, $\mathcal{B}$ repeats the process with another random pair (r, S) until finding a tuple (c, k', r) whose first two elements do not figure in a tuple of the list L_3. Before obtaining an admissible tuple (k', r, S), $\mathcal{B}$ must repeat the process at most $2q_R$ times (since L_3 can contain at most $2q_R$ entries). At each attempt, he must compute four pairings $\hat{e}$. Once an admissible tuple (k', r, S) is found, $\mathcal{B}$ puts (c, k', r) into L_3 before returning the ciphertext (c, r, S) which will appear to be valid from $\mathcal{A}$'s point of view.

If ID_A and ID_B are the identities ID_i and ID_j, $\mathcal{B}$ signcrypts M like this. He chooses $r^* \leftarrow_R \mathbb{F}_q^*$ and $S^* \leftarrow_R \mathbb{G}_1$, computes

$$k_1' = \hat{e}(P, S^*)\hat{e}(P_{pub}, Q_{ID_a})^{r^*} = \hat{e}(P, S^*)\hat{e}(cP, aP)^{r^*}$$

and chooses random $\tau^* \in_R \mathbb{G}_2$ and $k_2' \in_R \{0,1\}^n$ such that no entry $(., k_2')$ is in L_2 and comptutes $c^* = E_{k_2'}(M)$. He then verifies if the list L_3 already contains an entry (c^*, k_1', r') such that $r' \neq r^*$. If not, he puts the tuple (c^*, k_1', r^*) into L_3 and (τ^*, k_2') into L_2. In the opposite case, $\mathcal{B}$ chooses another random pair (r^*, S^*) and repeats the process as above until he finds a tuple (c^*, k_1', r^*) whose first two elements do not figure in an entry of L_3. Once he has admissible elements (r^*, S^*), $\mathcal{B}$ gives the ciphertext $\sigma^* = (c^*, r^*, S^*)$ to $\mathcal{A}$. $\mathcal{A}$ will never see that σ^* is not a valid signcrypted text of the plaintext M for identities ID_i and ID_j since he will not ask the unsigncryption of σ^*.

Unsigncrypt requests : When $\mathcal{A}$ observes a ciphertext $\sigma' = (c', r', S')$ for identities ID_i and ID_j, he may want to ask $\mathcal{B}$ for the unsigncryption of σ'. In such a case, $\mathcal{B}$ always notifies $\mathcal{A}$ that the ciphertext is invalid: if $\mathcal{A}$ previously asked the hash value $H_3(c', \hat{e}(P, S')\hat{e}(P_{pub}, Q_{ID_A})^{r'})$, there is a probability of at most $1/2^k$ that $\mathcal{B}$ answered r' (and that σ' was actually valid from $\mathcal{A}$'s point of view). The simulation fails if L_3 contains a tuple $(c', \hat{e}(P, S')\hat{e}(P_{pub}, Q_{ID_A})^{r'}, r')$. When receiving an unsigncryption query for a ciphertext $\sigma' = (c', r', S')$ for identities ID_A and ID_B that are not ID_i and ID_j, $\mathcal{B}$ first computes $k_1' = \hat{e}(P, S')\hat{e}(P_{pub}, Q_{ID_A})^{r'}$ and checks if the list L_3 contains the tuple (c', k_1', r'). If no such tuple is found, $\mathcal{B}$ rejects the ciphertext. Otherwise, he can recover r' and compute $\tau' = \hat{e}(S', Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^{r'}$ ($\mathcal{B}$ can simulate the knowledge of $\hat{e}(Q_{ID_A}, d_{ID_B})$ exactly as in the signcryption oracle simulation). He then searches for a query $H_2(\tau')$ in list L_2. If no such query is found, $\mathcal{B}$ takes a random pair $(\tau, k_2') \in \mathbb{G}_2 \times \{0,1\}^n$ such that no $(., k_2')$ already exists in L_2 and inserts (τ, k_2') into L_2. He finally uses the corresponding k_2' to find $m' = D_{k_2'}(c')$ and returns m'.

It is easy to see that, for all queries, the probability to reject a valid ciphertext does not exceed $q_U/2^k$. Indeed, for a query on identities ID_A and ID_B that are not ID_i and ID_j, if $\mathcal{A}$ later asks the hash value $H_3(c', \hat{e}(P, S')\hat{e}(P_{pub}, Q_{ID_A})^{r'})$, there is a probability of at most $1/2^k$ that $\mathcal{B}$ answers r' (and thus turns σ' into a valid ciphertext from $\mathcal{A}$'s point of view).

After a polynomially bounded number of queries, $\mathcal{A}$ chooses a pair of identities on which he wishes to be challenged. With a probability at least $1/\binom{q_{H_1}}{2}$, this pair of target identities will be (ID_i, ID_j) (we assume that after the first stage of the game, $\mathcal{A}$ chooses to be challenged on a pair of identities of which he asked the hashing). Notice that, if $\mathcal{A}$ asks the private key of ID_i or ID_j before choosing his target identities, then $\mathcal{B}$ fails because he is unable to answer the question (we recall that, if $\mathcal{A}$ actually chooses to be challenged on ID_i and ID_j, then he cannot ask ID_i nor ID_j's private keys in the second stage). If $\mathcal{A}$ does not choose ID_i and ID_j as target identities, then $\mathcal{B}$ fails.

When $\mathcal{A}$ produces his two plaintexts m_0 et m_1, $\mathcal{B}$ chooses a random bit $b \in_R \{0, 1\}$ and signcrypts m_b. To do so, he chooses $r^* \leftarrow_R \mathbb{F}_q^*$ and $S^* \leftarrow_R \mathbb{G}_1$. He computes

$$k_1' = \hat{e}(P, S^*)\hat{e}(P_{pub}, Q_{ID_a})^{r^*} = \hat{e}(P, S^*)\hat{e}(cP, aP)^{r^*},$$

$\tau^* = \hat{e}(S^*, Q_{ID_B})h^{r^*}$ (where h is $\mathcal{B}$'s candidate for the DBDH problem) to obtain $k_2' = H_2(\tau^*)$ (from the H_2 simulation algorithm) and $c_b = E_{k_2'}(m_b)$. He then verifies as above if L_3 already contains an entry (c_b, k_1', r') such that $r' \neq r^*$. If not, he puts the tuple (c_b, k_1', r^*) into L_3. In the opposite case, $\mathcal{B}$ chooses another random pair (r^*, S^*) and repeats the process until finding a tuple (c_b, k_1', r^*) whose first two elements do not figure in an entry of L_3. Once he has admissible elements (r^*, S^*), $\mathcal{B}$ just has to send the ciphertext $\sigma = (c_b, r^*, S^*)$ to $\mathcal{A}$.

$\mathcal{A}$ then performs a second series of queries which is treated in the same way as the first one. At the end of the simulation, he produces a bit b' for which he believes the relation $\sigma = \text{Signcrypt}(m_{b'}, d_{ID_i}, ID_j)$ holds. At this moment, if $b = b'$, $\mathcal{B}$ then answers 1 as a result because his candidate h allowed him to produce a σ that appeared to $\mathcal{A}$ as a valid signcrypted text of m_b. If $b \neq b'$, $\mathcal{B}$ then answers 0.

We now have to assess $\mathcal{B}$'s probability of success. We saw that $\mathcal{B}$ fails if $\mathcal{A}$ asks the private key associated to ID_i or ID_j during the first stage. We know that there are $\binom{q_{H_1}}{2}$ ways to choose the pair (ID_i, ID_j). Among those $\binom{q_{H_1}}{2}$ pairs of identities, at least one of them will never be the subject of a key extraction query from $\mathcal{A}$. Then, with a probability greater than $1/\binom{q_{H_1}}{2}$, $\mathcal{A}$ will not ask the questions Keygen(ID_i) and Keygen(ID_j). Further, with a probability exactly $1/\binom{q_{H_1}}{2}$, $\mathcal{A}$ chooses to be challenged on the pair (ID_i, ID_j) and this must allow $\mathcal{B}$ to solve his decisional problem if $\mathcal{A}$ wins the IND-IDSC-CCA game.

Finally, since $p_1 = P[b' = b|\sigma = \text{Signcrypt}(m_b, d_{ID_i}, ID_j)] = \frac{\epsilon+1}{2} - \frac{q_U}{2^k}$, and

$$p_0 = P[b' = i \mid h \in_R \mathbb{G}_2] = 1/2 \quad \text{for } i = 0, 1\ ,$$

we then have

$$\begin{aligned} Adv(\mathcal{B}) &= \left|P_{a,b,c \in_R \mathbb{F}_q}[1 \leftarrow \mathcal{B}(aP, bP, cP, \hat{e}(P, P)^{abc})] - P_{a,b,c \in_R \mathbb{F}_q, h \in_R \mathbb{G}_2}[1 \leftarrow \mathcal{B}(aP, bP, cP, h)]\right| \\ &= \frac{|p_1 - p_0|}{\binom{q_{H_1}}{2}^2} = \frac{\epsilon - q_U/2^{k-1}}{2\binom{q_{H_1}}{2}^2} > 2(\epsilon - q_U/2^{k-1})/q_{H_1}^4. \end{aligned}$$

We note that the denominator is $q_{H_1}^4$ rather than $q_{H_1}^2$ since $\mathcal{A}$ decides on which identities he wishes to be challenged after the first stage (it would have been $q_{H_2}^2$ if $\mathcal{A}$ could make his choice at the beginning of the game).

□

Proof of theorem 2

As in the proof of theorem 2, we use the attacker $\mathcal{A}$ to build a distinguisher $\mathcal{B}$ for the Decisional Bilinear Diffie-Hellman problem. $\mathcal{B}$ receives a random instance (P, aP, bP, cP, h) of the latter problem. He runs $\mathcal{A}$ with the system parameter $P_{pub} = cP$ and maintains lists L_1, L_2, L_3 to keep track of answers to hash queries. The identity hash queries H_1 are treated by $\mathcal{B}$ as in proof of theorem 1. The other queries are handled like this.

$\mathbf{H}_1$ queries : are treated as in proof of theorem 1. For random integers $i, j \in \{1, q_{H_1}\}$, $\mathcal{B}$ answers $H_1(ID_i) = aP$ and $H_1(ID_j) = bP$. He answers $H_1(ID_e) = dP$ and stores (ID_e, d) into L_1 for H_1 queries where $e \neq i, j$.

$\mathbf{H}_2$ queries : when receiving a query $H_2(g_e)$, $\mathcal{B}$ first checks if list L_2 already contains an entry (g_e, k_1'). If it does, $\mathcal{B}$ answers k_1'. Otherwise, he chooses a random $k_1 \in \mathbb{F}_q$ such that no entry $(., k_1')$ exists in L_2, puts (g_e, k_1) into L_2 and returns k_1 to $\mathcal{A}$.

$\mathbf{H}_3$ queries : are treated by $\mathcal{B}$ as H_2 queries using list L_3.

Key extraction queries : are treated exactly as in proof of theorem 1. $\mathcal{B}$ fails if $\mathcal{A}$ asks the private key corresponding to ID_i or ID_j.

Signcrypt queries : when receiving a signcryption query on (M, ID_A, ID_B), $\mathcal{B}$ checks if ID_A and ID_B are the identities ID_i, ID_j or not. If $ID_A \neq ID_i, ID_j$, $\mathcal{B}$ can extract the private key d_{ID_A} and run the signcryption algorithm Signcrypt(M, ID_A, ID_B).

If $ID_A = ID_i$ or $ID_A = ID_j$ and $ID_B \neq ID_i, ID_j$, $\mathcal{B}$ must simulate the operation like this. He runs the key extraction algorithm to obtain the private key d_{ID_B}. He then chooses random elements $(r, S) \in \mathbb{F}_q \times \mathbb{G}_1$ and computes $\tau_1 = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r$ and $\tau_2 = \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})$. The simulation made by $\mathcal{B}$ depend on whether list L_2 contains tuples of the form $(\tau_1, .)$ and $(\tau_2, .)$.

If L_2 contains entries (τ_2, k_2) and (τ_1, k_1) for some k_1, k_2 and if L_3 contains a tuple (M, k_2, u). If it does and if the first n bits of $r(k_1k_2)^{-1} \mod q$ differ from M, $\mathcal{B}$ chooses another pair (r, S) and repeats the process. If L_3 does not contain any entry like (M, k_2, u), $\mathcal{B}$ takes $u = [r(k_1k_2)^{-1} \mod q]_{n+1\ldots n+m}$ (where $[x]_{i\ldots j}$ denotes the bitstring between the i^{th} and j^{th} leftmost bits of x) and inserts the tuple (M, k_2, u) into L_3.

If there is an entry (τ_2, k_2) but no entry $(\tau_1, .)$, $\mathcal{B}$ checks if L_3 contains an entry (M, k_2, u). In this case, $\mathcal{B}$ computes $M' = M||u$ and $k_1 = (M'r^{-1}k_2)^{-1} \mod q$ before inserting (τ_1, k_1) into L_2. If no entry (M, k_2, u) exists in L_3, $\mathcal{B}$ picks a random $u \in \{0,1\}^m$, puts (M, k_2, u) into L_3 before computing $M' = M||u$ and k_1 as above in order to insert a "good" tuple (τ_1, k_1) into L_2.

In the case where no entry $(\tau_2, .)$ exists in L_2 but there is a (τ_1, k_1) for some k_1, $\mathcal{B}$ chooses a random $u \in \{0,1\}^m$ such that $(M, ., u)$ does not exist in L_3. He computes $M' = M||u$ and $k_2 = (M'r^{-1}k_1)^{-1}$. If no entry (M, k_2, u') with $u' \neq u$ exists in L_3, $\mathcal{B}$ puts (τ_2, k_2) into L_2 and (M, k_2, u) into L_3. Otherwise, $\mathcal{B}$ picks other elements (r, S) and repeats the process.

If no entries $(\tau_1, .)$ or $(\tau_2, .)$ exist in L_2, $\mathcal{B}$ can insert (τ_2, k_2) for a random k_2 and continue as above.

Once $\mathcal{B}$ has found admissible elements (r, S) and updated the lists, he returns (r, S) as the ciphertext corresponding to M. It is easy to see that the process must be repeated at most $2q_{H_3}$ times (since list L_3 contains at most $2q_{H_3}$ entries). At each attempt, four pairings must be computed.

For a signcryption query on a plaintext M and identities ID_i and ID_j, $\mathcal{B}$ returns a random ciphertext as in proof of theorem 3.1.

Unsigncrypt queries : When receiving an unsigncrypt query on a ciphertext (r, S) for identities ID_A and ID_B, $\mathcal{B}$ first computes $\tau_1 = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_A})^r$ and checks if L_2 contains a tuple of the form $(\tau_1, .)$. If not, he answers the ciphertext is invalid. Otherwise, $\mathcal{B}$ simulates the knowledge of $\hat{e}(Q_{ID_A}, d_{ID_B})$ as he does for signcryption queries. He computes $\tau_2 = \hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_A}, d_{ID_B})^r$ and checks if L_2 contains a tuple of the form $(\tau_2, .)$. If not, $\mathcal{B}$ rejects the ciphertext. Otherwise, $\mathcal{B}$ can recover $M' = r(k_1k_2)^{-1} \mod q$, take M_1 as the first n bits of M' and check if L_3 contains an entry $(M_1, k_2, .)$. If not, $\mathcal{B}$ rejects the ciphertext. Otherwise, he performs the validity checking as in the step 5 of the Unsigncrypt algorithm. Unsigncryption queries for identities ID_i and ID_j are treated as in proof of theorem 3.1. It is easy to see that the probability for $\mathcal{B}$ to reject a valid ciphertext is bounded by $q_U/2^k$ (where $k \approx \log_2 q$).

At the end of the first stage, $\mathcal{A}$ produces two plaintexts M_0 and M_1 and a pair of identities on which he wishes to be challenged. If the pair of identities is not (ID_i, ID_j), $\mathcal{B}$ fails and stops. Otherwise, he chooses a random bit b and signcrypts M_b using the candidate h instead of the pairing $\hat{e}(Q_{ID_i}, d_{ID_j})$. He might have to repeat the process several times before finding admissible (r, S). Once $\mathcal{B}$ has updated the lists L_1, L_2 and L_3, if h is actually the solution of the Bilinear Diffie-Hellman problem, $\sigma = (r, S)$ will appear as a valid signcryption to $\mathcal{A}$.

At the second stage, $\mathcal{A}$ performs a second series of queries that are treated exactly as in the first stage by $\mathcal{B}$. This time, $\mathcal{A}$ cannot ask the private keys corresponding to ID_i and ID_j.

At the end of the game, $\mathcal{A}$ outputs a bit b' for which he believes that the relation $\sigma = \text{Signcrypt}(M_{b'}, ID_i, ID_j)$. As in the previous proof, $\mathcal{B}$ outputs 1 if $b' = b$ and 0 otherwise. It is easy to see that $\mathcal{B}$'s probability of success is the same as in the proof of theorem 1.

$\square$