Identity-Based Hybrid Signcryption

Fagen Li$^{1,2,3}$, Masaaki Shirase$^3$, and Tsuyoshi Takagi$^3$

$^1$School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
$^2$Key Laboratory of Computer Networks and Information Security, Xidian University, Xi’an 710071, China
$^3$School of Systems Information Science, Future University-Hakodate, Hakodate 041-8655, Japan
Email: \{fagenli,shirase,takagi\}@fun.ac.jp

Abstract

Signcryption is a cryptographic primitive that fulfills both the functions of digital signature and public key encryption simultaneously, at a cost significantly lower than that required by the traditional signature-then-encryption approach. In this paper, we address a question whether it is possible to construct a hybrid signcryption scheme in identity-based setting. This question seems to have never been addressed in the literature. We answer the question positively in this paper. In particular, we extend the concept of signcryption key encapsulation mechanism to the identity-based setting. We show that an identity-based signcryption scheme can be constructed by combining an identity-based signcryption key encapsulation mechanism with a data encapsulation mechanism. We also give an example of identity-based signcryption key encapsulation mechanism.

1. Introduction

Identity-based (ID-based) cryptography was introduced by Shamir in 1984 [20]. The distinguishing property of ID-based cryptography is that a user’s public key can be any binary string, such as an email address that can identify the user. This removes the need for senders to look up the recipient’s public key before sending out an encrypted message. ID-based cryptography is supposed to provide a more convenient alternative to conventional public key infrastructure. Several practical ID-based signature schemes have been devised since 1984 but a satisfying ID-based encryption scheme only appeared in 2001 [7]. It was devised by Boneh and Franklin and cleverly uses bilinear maps (the Weil or Tate pairing) over supersingular elliptic curves.

Confidentiality, integrity, non-repudiation and authentication are the important requirements for many cryptographic applications. A traditional approach to achieve these requirements is to sign-then-encrypt the message. Signcryption, first proposed by Zheng [22], is a cryptographic primitive that fulfills both the functions of digital signature and public key encryption simultaneously, at a cost significantly lower than that required by the traditional signature-then-encryption approach. The original scheme in [22] is based on the discrete logarithm problem but no security proof is given. Zheng’s original construction [22] was only proven secure by Baek, Steinfeld, and Zheng [3] who described a formal security model in a multi-user setting. A recent direction is to merge the concepts of ID-based cryptography and signcryption to design efficient ID-based signcryption schemes. Several ID-based signcryption schemes have been proposed so far, e.g. [4], [8], [9], [10], [16], [18], [19].

The practical way to perform secrecy communication for large messages is to use hybrid encryption that separates the encryption into two parts: one part uses public key techniques to encrypt a one-time symmetric key; the other part uses the symmetric key to encrypt the actual message. In such a construction, the public key part of the algorithm is known as the key encapsulation mechanism (KEM) while the symmetric key part is known as the data encapsulation mechanism (DEM). A formal treatment of this paradigm originates in the work of Cramer and Shoup [11]. The resulting KEM-DEM hybrid encryption paradigm has received much attention in recent years [1], [14], [15]. It is very attractive as it gives a clear separation between the various parts of the cipher allowing for modular design. In [1], Abe, Gennaro, and Kurosawa introduced tag-KEM which takes as input a tag in KEM. Bentahar et al.’s [5] extended KEM into identity-based setting and proposed several efficient constructions of ID-based KEM (ID-KEM).

The use of hybrid techniques to build signcryption schemes has been studied by Dent [12], [13]. He generalized KEM to signcryption KEM which includes an authentication in KEM. However, he only consider the insider security for authenticity. That is, if the sender’s private key is exposed, an attacker is able to recover the key generated by signcryption KEM. The full insider security [2] means that (a) if the sender’s private key is exposed, an attacker is still not able to recover the message from the ciphertext and (b) if the receiver’s private key is exposed, an attacker is still not able to forge a ciphertext. In 2006, Bjørstad and Dent [6] showed how to build signcryption schemes using tag-KEM. However, they also only consider the insider security for authenticity and not for confidentiality. In 2008, Tan [21] proposed full insider secure signcryption KEM and tag-KEM without random oracles (in the standard model). Tan’s schemes are
insider secure for both authenticity and confidentiality.

All the above hybrid signcryption schemes [12], [13], [6], [21] is not ID-based. In this paper, we address a question whether it is possible to construct a hybrid signcryption scheme in ID-based setting. This question seems to have never been addressed in the literature. We answer the question positively in this paper. In particular, we extend the concept of signcryption KEM to the ID-based setting. We show that an ID-based signcryption scheme can be constructed by combining an ID-based signcryption KEM (IDSC-KEM) with a DEM. We also give an example of ID-based signcryption KEM. Our schemes are insider secure for both authenticity and confidentiality.

The rest of this paper is organized as follows. We introduce the preliminary work in Section 2. We give the formal model of ID-based signcryption KEM in Section 3. We show how to construct an ID-based signcryption scheme using an ID-based signcryption KEM and a DEM in Section 4. An example of ID-based signcryption KEM is described in Section 5. Finally, the conclusions are given in Section 6.

2. Preliminaries

2.1. ID-Based Signcryption (IDSC)

A generic ID-based signcryption scheme consists of the following four algorithms.

- **Setup**: is a probabilistic polynomial-time (PPT) algorithm run by a private key generator (PKG) that takes as input $1^k$ and outputs a master public key $mpk$ and a master secret key $msk$. Here $k$ is a security parameter.

- **Extract**: is a key generation algorithm run by the PKG that takes as input the master secret key $msk$ and an identity $ID \in \{0,1\}^*$, and outputs the corresponding private key $S_{ID}$.

- **Signcrypt**: is a PPT algorithm that takes as input a plaintext message $m$, a receiver’s identity $ID_r$, and a sender’s private key $S_{ID_s}$, and outputs a ciphertext $\sigma \leftarrow \text{Signcrypt}(m, S_{ID_s}, ID_r)$.

- **Unsigncrypt**: is a deterministic algorithm that takes as input a ciphertext $\sigma$, the receiver’s private key $S_{ID_r}$, and the sender’s identity $ID_s$, and outputs the original message $m$ or the symbol $\bot$ if $\sigma$ is an invalid ciphertext between identities $ID_s$ and $ID_r$.

We make the consistency constraint that if $\sigma \leftarrow \text{Signcrypt}(m, S_{ID_s}, ID_r)$, then $m \leftarrow \text{Unsigncrypt}(\sigma, S_{ID_r}, ID_s)$.

Malone-Lee [19] defines the security notions for ID-based signcryption schemes. These notions are semantic security (i.e. indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) and existential unforgeability against adaptive chosen messages attacks (UF-CMA)). For the stronger notion of insider security, we use the notion of strong existential unforgeability (sUF-CMA). The strong existential unforgeability means that an adversary wins if it outputs a valid message/signcryption pair $(m, \sigma)$ for identities $ID_s$ and $ID_r$ and the signcryption $\sigma$ was not returned by the signcryption oracle when queried on the message $m$.

For the confidentiality, we consider the following game played between a challenger $C$ and an adversary $A$.

- **Initial**: The challenger $C$ runs $(mpk, msk) \leftarrow \text{Setup}(1^k)$ and runs $A$ on input $(1^k, mpk)$.

- **Phase 1**: The adversary $A$ can perform a polynomially bounded number of queries in an adaptive manner (that is, each query may depend on the responses to the previous queries).

  - Key extraction queries: $A$ chooses an identity $ID$. $C$ computes $S_{ID} \leftarrow \text{Extract}(ID)$ and sends $S_{ID}$ to $A$.

  - Signcryption queries: $A$ produces a sender’s identity $ID_s$, a receiver’s identity $ID_r$, and a plaintext $m$. $C$ computes the private key $S_{ID_s} \leftarrow \text{Extract}(ID_s)$ and $\sigma \leftarrow \text{Signcrypt}(m, S_{ID_s}, ID_r)$ and sends $\sigma$ to $A$.

  - Unsighcrypt queries: $A$ selects a sender’s identity $ID_s$, a receiver’s identity $ID_r$, and a ciphertext $\sigma$. $C$ generates the private key $S_{ID_s} \leftarrow \text{Extract}(ID_s)$ and sends the result of $\text{Unsigncrypt}(\sigma, S_{ID_s}, ID_r)$ to $A$.

- **Challenge**: The adversary $A$ decides when Phase 1 ends. $A$ generates two equal length plaintexts $m_0, m_1$, a sender’s identity $ID_{s}^*$, and a receiver’s identity $ID_{r}^*$ on which it wishes to be challenged. The identity $ID_{s}^*$ should not appear in any key extraction queries in Phase 1. The challenger $C$ picks a random bit $\delta$ from $\{0,1\}$, computes $\sigma^* \leftarrow \text{Signcrypt}(m_{\delta}, S_{ID_{s}^*}, ID_{r}^*)$, and returns $\sigma^*$ to $A$.

- **Phase 2**: The adversary $A$ can ask a polynomially bounded number of queries adaptively again as in Phase 1 with the restriction that it cannot make a key extraction query on $ID_{r}^*$ and cannot make an unsighcrypt query on $\sigma^*$ to obtain the corresponding plaintext.

- **Guess**: The adversary $A$ produces a bit $\delta'$ and wins the game if $\delta' = \delta$.

The advantage of $A$ is defined to be

$$\text{Adv}_{\text{IND-CCA2}}^\text{IDSC}(A) = |2\Pr[\delta' = \delta] - 1|,$$


where $\Pr[\delta' = \delta]$ denotes the probability that $\delta' = \delta$.

**Definition 1**: An ID-based signcryption scheme is considered to be IND-CCA2 secure, if for all PPT adversaries $A$, the advantage in the IND-CCA2 game is a negligible function of the security parameter $k$.

Notice that the adversary is allowed to make a key extraction query on identity $ID_{s}^*$ in the above definition. This condition corresponds to the stringent requirement of insider security for confidentiality of signcryption [2]. On the other hand, it ensures the forward security of the scheme,
i.e. confidentiality is preserved in case the sender’s private key becomes compromised.

For the strong existential unforgeability, we consider the following game played between a challenger C and an adversary F.

• Initial: The challenger C runs \((mpk,msk) \leftarrow \text{Setup}(1^k)\) and runs F on input \((1^k,mpk)\).

• Attack: The adversary F performs a polynomially bounded number of queries just like in the confidentiality game.

• Forgery: F produces a quaternion \((m^*,\sigma^*,ID^*_s, ID^*_r)\), where the private key of \(ID^*_s\) was not asked and \(\sigma^*\) was not returned by the signcryption oracle on the input \((m^*,ID^*_s,ID^*_r)\) during Attack stage. F wins the game if the result of \(\text{Unsigncrypt}(\sigma^*,S_{ID^*_s}, ID^*_r)\) is not the \(\perp\) symbol.

The advantage of F is defined as the probability that it wins.

Definition 2: An ID-based signcryption scheme is considered to be sUF-CMA secure, if for all PPT adversaries F, the advantage in the sUF-CMA game is a negligible function of the security parameter \(k\).

Note that the adversary is allowed to make a key extraction query on the identity \(ID^*_r\) in the above definition. Again, this condition corresponds to the stringent requirement of insider security for authenticity of signcryption [2].

2.2. Date Encapsulation Mechanism (DEM)

A DEM is a symmetric encryption scheme which consists of the following two algorithms.

• Enc : is a deterministic encryption algorithm which takes as input \(1^k\), a key \(K\) and a message \(m \in \{0,1\}^*\), and outputs a ciphertext \(c \in \{0,1\}^*\), where \(K \in K_{DEM}\) is a key in the given key space, and \(m\) is a bit string of arbitrary length. We denote this as \(c \leftarrow \text{Enc}(K,m)\).

• Dec : is a deterministic decryption algorithm which takes as input a key \(K\) and a ciphertext \(c\), and outputs the message \(m \in \{0,1\}^*\) or a symbol \(\perp\) to indicate that the ciphertext is invalid.

For the purposes of this paper, it is only required that a DEM is secure with respect to indistinguishability against passive attackers (IND-PA). Formally, this security notion is captured by the following game played between a PPT adversary \(A\) and a challenger \(C\).

• Initial: \(A\) runs on input \(1^k\) and submits two equal length messages, \(m_0\) and \(m_1\).

• Challenge: \(C\) chooses a random key \(K \in K_{DEM}\) as well as a random bit \(\lambda \in \{0,1\}\), and sends \(c^* \leftarrow \text{Enc}(K,m_\lambda)\) to \(A\) as a challenge ciphertext.

• Guess: The adversary \(A\) produces a bit \(\lambda'\) and wins the game if \(\lambda' = \lambda\).

The advantage of \(A\) is defined to be

\[
\text{Adv}^{\text{IND-PA}}_{\text{DEM}}(A) = |2\Pr[\lambda' = \lambda] - 1|,
\]

where \(\Pr[\lambda' = \lambda]\) denotes the probability that \(\lambda' = \lambda\).

Definition 3: A DEM is considered to be IND-PA secure, if for all PPT adversaries \(A\), the advantage in the above game is a negligible function of the security parameter \(k\).

3. ID-Based Signcryption KEM

In this section, we give the formal definition for ID-based signcryption KEM.

3.1. Generic Scheme

A generic ID-based signcryption KEM consists of the following four algorithms.

• Setup : is a PPT algorithm which takes as input \(1^k\) and outputs the master public key \(mpk\) and the master secret key \(msk\). Here \(k\) is a security parameter.

• Extract : is a key generation algorithm which takes as input \(msk\) and an identity \(ID \in \{0,1\}^*\), and outputs the corresponding private key \(S_{ID}\).

• Encap : is a PPT key encapsulation algorithm which takes as input a plaintext message \(m\), a receiver’s identity \(ID_r\), and a sender’s private key \(S_{ID_s}\), and outputs an encapsulation key pair \((K,\psi)\), where \(K \in K_{\text{IDSC-KEM}}\) is a key in the space of possible session keys at a given security level, and \(\psi \in E_{\text{IDSC-KEM}}\) is the encapsulation of that key. We denote this as \((K,\psi) \leftarrow \text{Encap}(m,S_{ID_s},ID_r)\).

• Decap : is a deterministic key decapsulation algorithm which takes as input the sender’s identity \(ID_s\), the receiver’s private key \(S_{ID_r}\) and the encapsulation of that key \(\psi\), and outputs the corresponding key \(K\) or the error symbol \(\perp\). We denote this as \(K \leftarrow \text{Decap}(\psi,S_{ID_r},ID_s)\).

• Verify : is a deterministic verification algorithm which takes as input a sender’s identity \(ID_s\), a receiver’s private key \(S_{ID_r}\), a message \(m\), and an encapsulation \(\psi\), and outputs \(\top\) for “true” or \(\perp\) for “false”. Note that the verification algorithm does not need to take the symmetric key \(K\) as input as it can be easily computed from the encapsulation \(\psi\) using the deterministic decapsulation algorithm.

We make the consistency constraint that if

\[(K,\psi) \leftarrow \text{Encap}(m,S_{ID_s},ID_r),\]

then

\[K \leftarrow \text{Decap}(\psi,S_{ID_r},ID_s)\text{ and }\top \leftarrow \text{Verify}(\psi,m,S_{ID_r},ID_s)\]

3.2. Security Notions

An ID-based signcryption KEM should satisfy confidentiality and unforgeability. For the confidentiality, we consider the following game played between a challenger C and an adversary A.
• Initial: The challenger $C$ runs $(mpk,msk) \leftarrow \text{Setup}(1^k)$ and runs $A$ on input $(1^k, mpk)$.

• Phase1: The adversary $A$ can perform a polynomially bounded number of queries in an adaptive manner (that is, each query may depend on the responses to the previous queries).
  - Key extraction queries: $A$ chooses an identity $ID$. $C$ computes $S_{ID} \leftarrow \text{Extract}(ID)$ and sends $S_{ID}$ to $A$.
  - Key encapsulation queries: $A$ produces a sender’s identity $ID_s$, a receiver’s identity $ID_r$ and a plaintext $m$. $C$ computes the private key $S_{ID_s} \leftarrow \text{Extract}(ID_s)$ and $(K,\psi) \leftarrow \text{Encap}(m, S_{ID_s}, ID_r)$ and sends $(K,\psi)$ to $A$.
  - Key decapsulation queries: $A$ chooses a sender’s identity $ID_s$, a receiver’s identity $ID_r$, and an encapsulation $\psi$. $C$ generates the private key $S_{ID_s} \leftarrow \text{Extract}(ID_s)$ and sends the result of $\text{Decap}(\psi, S_{ID_s}, ID_r)$ to $A$.
  - Verification queries: $A$ chooses a sender’s identity $ID_s$, a receiver’s identity $ID_r$, a message $m$, and an encapsulation $\psi$. $C$ generates the private key $S_{ID_s} \leftarrow \text{Extract}(ID_s)$ and sends the result of $\text{Verify}(\psi, m, S_{ID_s}, ID_r)$ to $A$.

• Challenge: The adversary $A$ decides when Phase 1 ends. $A$ generates a message $m^*$, a sender’s identity $ID^*_s$, and a receiver’s identity $ID^*_r$ on which it wishes to be challenged. The identity $ID^*_s$ should not appear in any key extraction queries in Phase 1. $C$ then runs $(K_1, \psi^*) \leftarrow \text{Encap}(m, S_{ID^*_s}, ID^*_r)$ and randomly chooses $K_0 \leftarrow K_{\text{IDSC-KEM}}$. $C$ also chooses a random bit $b \in \{0,1\}$ and sends $(K_b, \psi^*)$ to $A$ as a challenge encapsulation key pair.

• Phase2: The adversary $A$ can ask a polynomially bounded number of queries adaptively again as in Phase 1 with the restriction that it cannot make a key extraction query on $ID^*_r$ and cannot make a decapsulation query on $(K_b, \psi^*)$ to obtain the corresponding key.

• Guess: The adversary $A$ produces a bit $b'$ and wins the game if $b' = b$.

The advantage of $A$ is defined to be

$$\text{Adv}_{\text{IDSC-KEM}}^{\text{IND-CCA2}}(A) = |2\Pr[b' = b] - 1|,$$

where $\Pr[b' = b]$ denotes the probability that $b' = b$.

Definition 4: An ID-based signcryption KEM is considered to be IND-CCA2 secure, if for all PPT adversaries $A$, the advantage in the IND-CCA2 game is a negligible function of the security parameter $k$.

For the unforgeability, we consider the following game played between a challenger $C$ and an adversary $F$.

• Initial: The challenger $C$ runs $(mpk,msk) \leftarrow \text{Setup}(1^k)$ and runs $F$ on input $(1^k, mpk)$.

• Attack: The adversary $F$ performs a polynomially bounded number of queries just like in the confidentiality game.

• Forgery: $F$ produces a quaternion $(m^*, \psi^*, ID^*_s, ID^*_r)$, where the private key of $ID^*_s$ was not asked and $\psi^*$ was not returned by the key encapsulation oracle on the input $(m^*, ID^*_s, ID^*_r)$ during attack stage. $F$ wins the game if the result of $\text{Verify}(\psi^*, m^*, S_{ID^*_s}, ID^*_r)$ is not the symbol $\bot$.

The advantage of $F$ is defined as the probability that it wins.

Definition 5: An ID-based signcryption KEM is considered to be sUF-CMA secure, if for all PPT adversaries $F$, the advantage in the sUF-CMA game is a negligible function of the security parameter $k$.

4. Identity-Based Hybrid Signcryption

We can combine an ID-based signcryption KEM with a DEM to form an ID-based hybrid signcryption scheme. We describe it in Figure 1.

Setup: On input $1^k$:
1. $(mpk,msk) \leftarrow \text{IDSC-KEM.Setup}(1^k)$
2. Output the master public key $mpk$ and the master secret key $msk$

Extract: On input the master secret key $msk$ and an identity $ID \in \{0,1\}^*$:
1. $S_{ID} \leftarrow \text{IDSC-KEM.Extract}(msk, ID)$
2. Output the private key $S_{ID}$ of the identity $ID$

Signcrypt: On input the sender’s private key $S_{ID_s}$, the receiver’s identity $ID_r$, and a message $m \in \{0,1\}^*$:
1. $(K,\psi) \leftarrow \text{IDSC-KEM.Encap}(m, S_{ID_s}, ID_r)$
2. $c \leftarrow \text{DEM.Enc}(K, m)$
3. Output the ciphertext $\sigma \leftarrow (\psi, c)$

Unsigncrypt: On input the sender’s identity $ID_s$, the receiver private key $S_{ID_r}$ and the ciphertext $\sigma$:
1. $K \leftarrow \text{IDSC-KEM.Decap}(\psi, S_{ID_r}, ID_s)$
2. If $K = \bot$, then output $\bot$ and stop
3. $m \leftarrow \text{DEM.Dec}(K)$
4. If $\bot \leftarrow \text{IDSC-KEM.Verify}(\psi, m, S_{ID_s}, ID_r)$, output $m$. Otherwise output $\bot$.

Figure 1. Identity-based hybrid signcryption

We give the security results for ID-based hybrid signcryption in Theorems 1 and 2.
Theorem 1: Let IDSC be an ID-based hybrid signcryption scheme constructed from an ID-based signcryption KEM and a DEM. If the ID-based signcryption KEM is IND-CCA2 secure and the DEM is IND-PA secure, then IDSC is IND-CCA2 secure. In particular, we have

\[ \text{Adv}_{\text{IDSC}}^{\text{IND-CCA2}}(A) \leq 2 \text{Adv}_{\text{IDSC-KEM}}^{\text{IND-CCA2}}(B_1) + \text{Adv}_{\text{DEM}}^{\text{IND-PA}}(B_2). \]

Proof: See the full paper [17].

Theorem 2: Let IDSC be an ID-based hybrid signcryption scheme constructed from an ID-based signcryption KEM and a DEM. If the ID-based signcryption KEM is sUF-CMA secure, then IDSC is sUF-CMA secure. In particular, we have

\[ \text{Adv}_{\text{IDSC}}^{\text{sUF-CMA}}(F) \leq \text{Adv}_{\text{IDSC-KEM}}^{\text{sUF-CMA}}(F), \]

where \( \text{Adv}_{\text{IDSC}}^{\text{sUF-CMA}}(F) \) is the advantage of the sUF-CMA adversary against IDSC, and \( \text{Adv}_{\text{IDSC-KEM}}^{\text{sUF-CMA}}(F) \) is the advantage of the resulting sUF-CMA adversary against ID-based signcryption KEM.

Proof: See the full paper [17].

5. An Example of ID-Based Signcryption KEM

Most of ID-based signcryption schemes [4], [8], [9], [10], [16], [18], [19] fit the new generic framework. Here we give an example of ID-based signcryption KEM based on Barreto et al.’s scheme [4]. Barreto et al.’s scheme is the fastest ID-based signcryption scheme so far. If we combine the ID-based signcryption KEM with a DEM as Figure 1, we can get a scheme that is very similar to Barreto et al.’s original scheme. Since Barreto et al.’s scheme uses the bilinear pairings, we first describe the basic definition and properties of the bilinear pairings.

5.1. Bilinear Pairings

Let \( G_1, G_2 \) and \( G_T \) be three cyclic groups of prime order \( q \). Let \( P, Q \) be generators of \( G_1 \) and \( G_2 \), respectively. A bilinear pairing is a map \( \hat{e} : G_1 \times G_2 \rightarrow G_T \) with the following properties:

1) Bilinearity: \( \forall (S, T) \in G_1 \times G_2, \forall a, b \in \mathbb{Z}_q, \hat{e}(aS, bT) = \hat{e}(S, T)^{ab}. \)
2) Non-degeneracy: \( \forall S \in G_1, \hat{e}(S, T) = 1 \) for all \( T \in G_2 \) iff \( S = 0 \).
3) Computability: \( \forall (S, T) \in G_1 \times G_2, \hat{e}(S, T) \) is efficiently computable.
4) There exists an efficient, publicly computable (but not necessarily invertible) isomorphism \( \varphi : G_2 \rightarrow G_1 \) such that \( \varphi(Q) = P \).

The security of Barreto et al.’s scheme relies on the hardness of the following problems.

Definition 6: Define \( G_1, G_2, G_T \) and \( \hat{e} \) as in this section. The l-Strong Diffie-Hellman problem \( (l, \text{SDHP}) \) in the groups \( (G_1, G_2) \) is to find a pair \( (c, \frac{1}{c}P) \) with \( c \in \mathbb{Z}_q^* \) given a \( (l + 2) \)-tuple \( (P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^l Q) \).

Definition 7: Define \( G_1, G_2, G_T \) and \( \hat{e} \) as in this section. The l-Bilinear Diffie-Hellman Inversion problem \( (l-BDHIP) \) in the groups \( (G_1, G_2, G_T) \) is to compute \( \hat{e}(P, Q)^{1/n} \in G_T \) given \( (P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^l Q) \).

5.2. ID-Based Signcryption KEM

- Setup: Define \( G_1, G_2, G_T \) and \( \hat{e} \) as in previous subsection. Let \( H_1, H_2 \) and \( H_3 \) be three cryptographic hash functions where \( H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*, H_2 : \{0, 1\}^* \times G_T \rightarrow \{0, 1\}^n \). Here \( n \) is the key length of a DEM. Let \( Q \in G_2, P = \varphi(Q) \in G_1 \) be generators of \( G_2 \) and \( G_1 \), respectively and \( g = \hat{e}(P, Q) \in G_T \). The PKG chooses a master secret key \( s \in \mathbb{Z}_q^* \) randomly and computes \( Q_{pub} = sQ \in G_2 \). The PKG publishes system parameters \{\( G_1, G_2, G_T, P, Q, g, Q_{pub}, \hat{e}, \varphi, H_1, H_2, H_3 \}\} and keeps the master key secret.
- Extract: Given an identity \( ID \), the PKG computes the private key \( S_{ID} \leftarrow \frac{1}{\hat{e}(H_1(ID), P + \varphi(Q_{pub}))}. \)
- Encap: Given a message \( m \), a receiver’s identity \( ID_r \) and a sender’s private key \( S_{ID_s} \), this algorithm works as follows.
  1) Choose \( x \in \mathbb{Z}_q^* \) randomly and compute \( r \leftarrow g^x \).
  2) Compute \( K \leftarrow H_3(r) \).
  3) Compute \( h \leftarrow H_2(m, r) \).
  4) Compute \( S \leftarrow (x + h) \varphi(S_{ID_s}) \).
  5) Compute \( T \leftarrow x(H_1(ID_r)P + \varphi(Q_{pub})). \)
  6) Set \( \psi \leftarrow (S, T) \).
  7) Output \((K, \psi)\).
- Decap: Given the sender’s identity \( ID_s \), the receiver’s private key \( S_{ID_r} \), and an encapsulation \( \psi \), this algorithm works as follows.
  1) Compute \( r \leftarrow \hat{e}(T, S_{ID_r}) \).
  2) Compute \( K \leftarrow H_3(r) \).
  3) Output \( K \).
- Verify: Given the sender’s identity \( ID_s \), the receiver’s private key \( S_{ID_r} \), a message \( m \), and an encapsulation \( \psi \), this algorithm works as follows.
  1) Compute \( r \leftarrow \hat{e}(T, S_{ID_r}) \).
  2) Compute \( h \leftarrow H_2(m, r) \).
  3) If \( r = \hat{e}(S, H_1(ID_r)Q + Q_{pub})g^{-h} \), output symbol \( \top \). Otherwise, output symbol \( \bot \).

In a real implementation of this algorithm, we can store the value of \( r \) computed by the decapsulation algorithm and use it again in the verification algorithm. Such an implementation would be functionally identical to the above algorithm and would therefore be just as secure. We choose to separate the decapsulation and verification algorithms so that they can be studied independently. The security proof is similar to that of [4]. We omit it.
6. Conclusions

In this paper, we extended the concept of signcryption KEM to the identity-based setting. We showed that an ID-based signcryption scheme can be constructed by combining an ID-based signcryption KEM with a DEM. To show that our framework is reasonable, we also gave an example of ID-based signcryption KEM based on Barreto et al.'s ID-based signcryption scheme.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 60803133), the Key Laboratory of Computer Networks and Information Security of Xidian University (Grant No. 2008CNSIS-02), and Youth Science and Technology Foundation of UESTC. Fagen Li is supported by the JSPS postdoctoral fellowship for research in Japan.

References


