2009 Fourth International Conference on Computer Sciences and Convergence Information Technology

Identity-Based Secret Signature Scheme

Byoungcheon Lee
Dept. of Information Security
Joongbu University
Geumsan-Gun, Chungnam, Korea
sultan@joongbu.ac.kr

Jin Li
Department of ECE
Illinois Institute of Technology
Chicago, Illinois 60616, USA
jli25@iit.edu

Kwangjo Kim
School of Engineering
KAIST
Daejeon, Korea
kkj@kaist.ac.kr

***Abstract*—Secret signature, proposed by Lee *et al.* [11], is a new signature scheme which provides signature privacy and public provability together. Using the secret signature scheme a signer can send his signature secretly to a specific receiver such that only the designated receiver can verify the signature. If any argument occurs between them, the validity of the secret signature can be proven publicly either by the signer or the receiver. Thus it can be used as an important cryptographic primitive to implement private business transactions.**

In this paper we present an identity-based implementation of secret signature scheme which can enjoy the advantage of reducing key management load compared with the traditional certificate-based cryptography. We modify Cha-Cheon's ID-based signature scheme [3] and ID-based key agreement scheme [14] to provide public provability, and then combine them to implement ID-based secret signature (IBSS) scheme.

***Keywords*-Secret signature; Signature privacy; Unforgeability; Invisibility; Receiver designation; Identity-based cryptography;**

I. Introduction

Digital signature, first proposed by Diffie and Hellman [7] in 1976, is an electronic version of handwritten signatures for digital documents. A digital signature, σ, on a message, m, is generated by a signer, S, using a secret signing key, sk_S. The correctness of the generated signature is verified using the corresponding public key, pk_S, of the signer. It provides authentication and non-repudiation in a public way, since it can be verified by anyone (public verifiability). In the research community there had been extensive effort to design signature variant schemes which can provide some additional properties required in various applications. In this paper we consider the signature privacy issue.

Signature Privacy Issue. Digital signatures have been used for many public transactions to achieve public verifiability, but they are also frequently used for private business transactions. Let's consider a scenario where both a sender (signer) and a receiver (verifier) wish to keep their exchanged signatures private – we term this *signature privacy*. We think that signature privacy has to be one of the basic security requirement of digital signature.

A 'straightforward' approach to achieve signature privacy is to encrypt a digital signature with the receiver's public key (or with an agreed key) so that only the legitimate receiver can decrypt and retrieve the original signature. This is the so-called *sign-then-encrypt* approach, which is widely adopted in the real world. Several schemes proposed in the literature can also be used to achieve signature privacy, for example, signcryption [15], designated verifier signature (DVS) [4], [10] and limited verifier signature (LVS) [1], [6]. Recently, Lee *et al.* [11] proposed a new notion of digital signature, called *secret signature*, which provides authentication and non-repudiation secretly only to a designated receiver.

Assume that we are involved in a private business transaction and use a signature scheme designed to provide *signature privacy*. In the event that a dispute arises between a signer and a receiver after private transaction, they may want to resolve the dispute by proving the validity of the private signature to third party (e.g., a judge or the public). We term such a requirement the *public provability* of a signature. If we use general digital signature schemes, this property is easily achieved by verifying the signature using the signer's public key. If we use signature schemes designed to provide signature privacy, then the public provability of signature becomes an important requirement to ensure the fairness of business transactions.

Secret Signature Scheme. In this section we review the secret signature scheme by Lee *et al.* [11]. The basic idea of secret signature is combining a signature scheme and a non-interactive one-way key agreement scheme. To generate a secret signature a signer first generates an agreed key with a designated-receiver using a non-interactive one-way key agreement technique and then signs message and the agreed key together. Upon receiving the secret signature the designated receiver can compute the agreed key and then verify the secret signature. But any other entity cannot verify the signature because of the key agreement part. Thus the secret signature is a completely private transaction between the signer and the designated receiver. More detailed discussion on its properties compared with other signature privacy-related signature schemes is given in [11].

Secret signature is an important cryptographic primitive which can be used to implement private transaction between two entities (exchange signatures privately and secretly). We consider an application to the delivery of receipt. If a customer buys a product with a payment, the merchant will

978-0-7695-3896-9/09 $26.00 © 2009 IEEE
DOI 10.1109/ICCIT.2009.182

issue a receipt. Basically this kind of interaction is a private business which has no need to reveal to others. In this case the merchant can issue a receipt using the secret signature scheme with the customer as a designated receiver. If any dispute arises afterward (after service, refund, warrant), both the customer and the merchant can reveal the receipt.

Our Contribution. Identity-based cryptosystem [13] is a public key cryptosystem where the public key can be any arbitrary string such as identity, email address, *etc*. The corresponding private key is generated by a trusted entity called key generation center (KGC) using a master secret key and is sent to the user through a secure channel. Identity-based cryptosystem can provide great flexibility in key management since key distribution and certification are not required. An identity-based encryption scheme allows a sender to send an encrypted message to a receiver without accessing the public key certificate of the receiver. Boneh and Franklin [2] proposed the first practical identity-based encryption scheme based on bilinear maps on elliptic curves. Subsequently, identity-based signature scheme [3] was proposed.

So, it is a challenging task to provide secret signature scheme in ID-based setting. The straightforward approach is combining existing ID-based signature schemes [3] and ID-based key agreement schemes [14], [5], but it looks hard to provide public provability in this basic approach. In this paper we modify ID-based signature scheme [3] and ID-based key agreement scheme [14] to provide public provability, and combine them to implement ID-based secret signature (IBSS) scheme. We first give the security definition and security model for IBSS. Then, we show how to construct a secure IBSS scheme by using pairing. Under the given security definition and security model the proposed scheme is proven secure in the random oracle model under the standard computational Diffie-Hellman (CDH) and decisional Diffie-Hellman (DDH) assumptions.

We formally define the proposed IBSS scheme and provide security definitions in Section 2. We then present the proposed IBSS scheme and prove its security under the given security model in Section 3. We describe two public proof protocols, general proof and anonymous proof, in Section 4. Finally, we conclude the paper in Section 5.

II. Definitions

A. Definition of Identity-Based Secret Signature Scheme

There are three entities in the IBSS scheme, namely, key generation center (KGC), signer (sender) S, and verifier (receiver) R. The formal definition of IBSS scheme is as follows.

Definition 1: Identity-Based Secret Signature Scheme. An identity-based secret signature (IBSS) scheme consists of the following four algorithms:

1) **Setup:** $(\mathsf{params}, pk, sk) \leftarrow Setup(1^k)$
   It is a probabilistic algorithm that takes a security parameter k as input and outputs the public parameters params and PKG's key pair (pk, sk).
2) **Key Generation:** $sk_{ID} \leftarrow KG(ID, \mathsf{params})$
   It is a probabilistic algorithm that takes identity ID and the public parameters params as input, and outputs sk_{ID}, the secret key for ID.
3) **Signing:** $(\sigma, \mathsf{seed}) \leftarrow S(\mathsf{params}, m, sk_{ID_S}, ID_R)$
   It is a probabilistic algorithm that takes as input the public parameters params, a plaintext message $m \in \{0,1\}^*$, signer's private key sk_{ID_S}, receiver's identity ID_R, and outputs a secret signature σ and seed.
4) **Verification:** $\mathsf{result} \leftarrow V(\mathsf{params}, \sigma, m, ID_S, sk_{ID_R})$
   It is a deterministic algorithm that takes as input the public parameters params, a secret signature σ, a plaintext message m, signer's identity ID_S, receiver's private key sk_{ID_R}, and outputs result. If σ is a valid secret signature, then result = valid; otherwise, result = invalid.

IBSS scheme is a private transaction between a signer and a designated receiver, and it is hidden from others. But if any dispute arises between them, they may want to prove its validity to others to resolve the dispute. Public proving of secret signature can be divided into two sub-algorithms; *signature proving* to prove the validity of secret signature and *receiver proving* to prove the identity of the receiver. Signature proving is used to show that the secret signature was generated by the signer without revealing who is the receiver. On the other hand, receiver proving is used to show that the secret signature was designated to the receiver. These public proving schemes can be done either by the signer or the receiver.

Definition 2: Signature Proving Scheme. A signature proving scheme consists of the following two algorithms.

1) **Signature proof:** $w \leftarrow SP(\cdot)$
   It is a deterministic algorithm that computes the agreed key w used in the signing stage and outputs it. Computation of w is different depending on whether the prover is the signer or the receiver.
2) **Signature proof verification:**
   $\mathsf{result} \leftarrow SPV(\mathsf{params}, \sigma, m, w, ID_S)$
   It is a deterministic algorithm that takes as input params, a secret signature σ, a message m, an agreed key information w, the identities of the signer, and outputs a verification result, either valid or invalid.

Definition 3: Receiver Proving Scheme. A receiver proving scheme consists of the following two algorithms.

1) **Receiver proof** algorithm can be done either by the signer or the receiver.
   - **Receiver proof by signer:**
     $\mathsf{proof_S} \leftarrow RPS(\mathsf{params}, w, \mathsf{seed})$
     It is a probabilistic algorithm that takes as input params, agreed key w, random seed used by the

signer in the signing stage, and outputs $\mathsf{proof_S}$.

- **Receiver proof by receiver:**
  $\mathsf{proof_R} \leftarrow RPR(\mathsf{params}, w, sk_{ID_R})$
  It is a probabilistic algorithm that takes as input params, agreed key w, the receiver's private key sk_{ID_R}, and outputs $\mathsf{proof_R}$.

2) **Receiver Verification:**
   $\mathsf{result} \leftarrow RV(\mathsf{params}, w, ID_R, \mathsf{proof})$
   It is a deterministic algorithm that takes as input params, an agreed key information w, the identity of the receiver, receiver proof (either $\mathsf{proof_S}$ or $\mathsf{proof_R}$), and outputs a verification result, either valid or $\mathsf{invalid}$.

B. Security Definitions

To be used for private business transaction we define the security requirements of secret signature by unforgeability (non-repudiation), invisibility (signature privacy), receiver designation, and public provability. Here we give some intuition for these security requirements.

- Unforgeability: Anyone except the signer can have a non-negligible advantage in forging a secret signature. If unforgeability is provided, then the non-repudiation of signer is obtained consequently.
- Invisibility: The secret signature generated by the signer is verifiable only by the designated receiver. No other entity except the signer or the receiver is able to have a non-negligible advantage in distinguishing the secret signature. Signature privacy is defined in terms of invisibility.
- Receiver designation: A secret signature is given to a designated receiver. If a need arises, the identity of designated receiver can be revealed. Therefore any other entity is completely out of business for the secret signature.
- Public provability: The validity of the secret signature and the identity of receiver can be proven to others either by the signer or the verifier, if the need arises.

For unforgeability and invisibility we give more formal definitions in the following.

To define the unforgeability more formally, we recall the widely accepted security notions on digital signatures. A formalized and widely accepted security notion for digital signature was introduced by Goldwasser, Micali, and Rivest [9], which they term *existential unforgeability under adaptive chosen-message attack* (EF-ACMA).

However, in our proposed IBSS scheme, there are two inputs: the message to be signed and the intended receiver's public key. Hence, we extend the standard security definition to the *existential unforgeability under the adaptive chosen-message chosen-receiver attack* (EF-ACMCRA) in which the attacker is allowed to query secret signatures to the signing oracle for adaptively chosen messages and receivers. An unforgeability of IBSS can be defined in terms of the following unforgeability game.

Game Unforgeability: Let $\mathcal{F}$ be a forger and k be a security parameter. We assume the existence of signing oracle and key generation oracle.

1) $(\mathsf{params}, pk, sk) \leftarrow Setup(1^k)$ is executed to get the KGC's key pair. $sk_{ID_S} \leftarrow KG(ID_S, \mathsf{params})$ is executed to generate signer's private key. (params, pk) is given to $\mathcal{F}$ and sk_{ID_S} is given to the signing oracle.
2) $\mathcal{F}$ is allowed to ask a series of key generation queries to the key generation oracle for any receiver ID_R (except the signer ID_S) to get the receiver's private key sk_{ID_R}. $\mathcal{F}$ is also allowed to ask a series of signing queries to the signing oracle which provides secret signatures of the signer ID_S. $\mathcal{F}$ can ask secret signatures for any combination of m and ID_R adaptively chosen by $\mathcal{F}$. It asks secret signature to the signing oracle by sending (m, ID_R), then the signing oracle provides valid secret signatures σ.
3) Finally $\mathcal{F}$ outputs a pair (m^*, ID_R^*, σ^*) as a forged secret signature σ^* of the signer ID_S on message m^* given to the receiver ID_R^*.

If $\mathsf{valid} \leftarrow Verify(\mathsf{params}, \sigma^*, m^*, ID_S, sk_{ID_R^*})$ and the tuple (m^*, ID_R^*) has never been queried to the signing oracle, then $\mathcal{F}$ wins the game. In this definition of unforgeability we allow that receiver's private key is given to $\mathcal{F}$ and the signing oracle. Without the knowledge of receiver's private key the signing oracle cannot simulate secret signature and $\mathcal{F}$ cannot verify the validity of received secret signature. Since the main concern of the unforgeability game is the unforgeability of signer's signing ability, this model is reasonable.

Definition 4: Unforgeability. An IBSS scheme is defined to be secure in the sense of existential unforgeability under the adaptive chosen-message chosen-receiver attack (EF-ACMCRA), if no probabilistic polynomial-time (PPT) forger, $\mathcal{F}$, can have a non-negligible advantage in the above game unforgeability.

Signature privacy requires that a secret signature is a private information between the signer and the receiver, and it is invisible to any other entity. Signature privacy is defined in terms of the following invisibility game.

Game Invisibility: Let $\mathcal{D}$ be a distinguisher and $\mathcal{C}$ be a challenger. Signer ID_S and receiver ID_R is fixed in this game. $\mathcal{D}$ is a third party who is neither the signer nor the receiver.

1) $(\mathsf{params}, pk, sk) \leftarrow Setup(1^k)$ is executed to get the PKG's key pair. $sk_{ID_S} \leftarrow KG(ID_S, \mathsf{params})$ is executed to generate signer's private key. (params, pk) is given to $\mathcal{D}$ and sk_{ID_S} is given to the signing oracle.
2) At some point $\mathcal{D}$ outputs two challenge messages m_0 and m_1 and requests for a challenge secret signature σ to the challenger $\mathcal{C}$.
3) The challenger $\mathcal{C}$ makes a hidden coin toss b and computes σ for message m_b by using the signing

oracle. σ is given to $\mathcal{D}$.

4) The distinguisher $\mathcal{D}$ is allowed to ask a series of key generation queries except ID_S and ID_R. $\mathcal{D}$ is also allowed to ask a series of signing queries for any message m' (except m_0 and m_1) and receiver.
5) Finally, $\mathcal{D}$ outputs a guess b'.

$\mathcal{D}$ wins the invisibility game if $b = b'$.

Definition 5: Invisibility. An IBSS scheme is said to be secure in the sense of invisibility, if no probabilistic polynomial-time distinguisher, $\mathcal{D}$, can have a non-negligible advantage in the above game.

If a secret signature scheme is invisible, it is hidden from any other entities except the signer and the receiver. Thus it provides signature privacy.

III. Our Construction of IBSS

Let G_1 and G_2 be cyclic groups of prime order q with the multiplicative group action. Let g be a generator of G_1 and $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a bilinear map with the following properties:

1) Bilinearity: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $g_1, g_2 \in G_1$, and $a, b \in_R Z_q$;
2) Non-degeneracy: There exists $g_1, g_2 \in G_1$ such that $\hat{e}(g_1, g_2) \neq 1$, in other words, the map does not send all pairs in $G_1 \times G_1$ to the identity in G_2;
3) Computability: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in G_1$.

As shown in [11] secret signature can be constructed as a combination of secure signature scheme and key agreement scheme. The straightforward approach is combining existing ID-based signature schemes [3] and ID-based key agreement schemes [5], [14], but it looks hard to provide public provability.

In this paper we modify [3] and [14] to provide public provability, and then combine them to implement ID-based secret signature (IBSS) scheme. The proposed ID-based secret signature scheme consists of the following 4 algorithms.

1. Setup: Define two secure cryptographic hash functions, $H : \{0,1\} \rightarrow G_1$ and $H' : \{0,1\} \rightarrow Z_q$. Choose a random number $x \in Z_q^*$ and computes $pk = g^x$. pk is a public key of KGC and x is the master key of KGC.

2. Key Generation: KGC issues private keys to entities through a secure channel. Signer S gets a private key $sk_{ID_S} = pk_{ID_S}^x$ where $pk_{ID_S} = H(ID_S)$. Receiver R gets a private key $sk_{ID_R} = pk_{ID_R}^x$ where $pk_{ID_R} = H(ID_R)$.

3. Signing: Let m denote the message to be signed. The signer S, chooses two random numbers $k, k_1 \in_R Z_q^*$ and computes $u = H(ID_S)^k, f = g^{k_1}$, and $w = \hat{e}(pk, H(ID_R)^{k_1})$. Here w is a secret shared key with the receiver. Finally, it computes $v = sk_{ID_S}^{k+H'(m,ID_S,u,f,w)}$. It sends $\sigma = (u, f, v)$ to the intended receiver R as a secret signature on message m. Note that σ can be considered as a Cha-Cheon's signature on message m and agreed key w.

4. Verification: The receiver R retrieves the shared key w using its private key sk_{ID_R} by $w = \hat{e}(f, sk_{ID_R})$. It then verifies $\sigma = (u, f, v)$ by

$$\hat{e}(v, g) \stackrel{?}{=} \hat{e}(pk, u \cdot H(ID_S)^{H'(m,ID_S,u,f,w)}).$$

If the above verification is valid, R receives σ as a valid secret signature of S on message m.

Secret signature is a private transaction between a signer and a designated receiver, and it is hidden from others. $\sigma = (u, f, v)$ is a hidden signature of the signer S for a message m given to the designated receiver R. Here we do not focus on the secrecy of the message. Message can be sent in plaintext, in ciphertext using other encryption scheme, or shared in advance. If some attackers observe the transaction between the signer and receiver, they may be able to guess the contents of the transaction, but they cannot have clear evidence on private interactions.

We prove the security of the proposed scheme in the sense of unforgeability.

Theorem 1: The proposed IBSS scheme is EF-ACMCRA secure in the sense of definition 4 in the random oracle model, under the assumption that the CDH assumption holds.

Proof sketch. Suppose that there exists a forger $\mathcal{F}$ who breaks the secret signature scheme in an existential unforgeability under the adaptive chosen-message and chosen-receiver attack. Assume it can forge a secret signature in time $t(k)$ with a non-negligible advantage $\epsilon(k)$. In the training stage $\mathcal{F}$ can ask signing queries for any combination of message and receiver pair to the signing oracle and receive correct secret signatures from the signing oracle. The challenger $\mathcal{C}$ controls all communication of $\mathcal{F}$ and simulate the signing queries. $\mathcal{C}$ described below will solve the CDH problem; for a randomly given instance $\{g, X = g^x, Y = g^y\}$, compute g^{xy}.

1) Simulation of Setup: Let $pk = X$ as the public key of KGC.
2) Simulation of Random Oracle: Assume $\mathcal{F}$ makes at most q_H queries to the H-oracle. $\mathcal{C}$ randomly chooses a number $s \in [1, q_H]$. When $\mathcal{F}$ queries ID_i to H-oracle and $i \neq s$, $\mathcal{C}$ answers $H(ID_i) = g^{r_i}$ for a random $r_i \in Z_p$. When $i = s$, $\mathcal{C}$ answers $H(ID_s) = Y$.
3) Simulation of Key Generation: Assume the adversary asks for a secret key for identity ID. $\mathcal{C}$ first checks if there is any query for H(ID) in the list of H-query. If there is such query and $i \neq s$, $\mathcal{C}$ simulates and answers $sk_{ID} = X^{r_i}$. If there is no such query in H-query list, then $\mathcal{C}$ sets $H(ID) = g^{r_i}$ and answers the private key extraction by $sk_{ID} = X^{r_i}$.
4) Simulation of Signing Oracle: If the signing query is for $ID \neq ID_s$, then, $\mathcal{C}$ just generates the signature like the simulation of signing algorithm in [3].

Finally, the forger $\mathcal{F}$ outputs a forged signature $\sigma = (u, f, v)$ of the signer ID_S to the receiver ID_R on message m. Let's denote $h = H'(m, ID_S, u, f, w)$ in the H' oracle. By using the forking lemma, $\mathcal{C}$ can get another valid signature $\sigma' = (u, f', v')$ with $h' = H'(m, ID_S, u, f', w')$. Then $\mathcal{C}$ solves the CDH problem by sk_{ID_S}=$(v/v')^{(h-h')^{-1}}$=g^{xy}.

IV. Public Proving of Secret Signature

Public proving of secret signature has two sub-algorithms; signature proving to prove the validity of secret signature and receiver proving to prove the identity of the receiver. Signature proving is used to show that the secret signature generated by the signer is valid without revealing who is the receiver. On the other hand, receiver proving is used to show that the secret signature is designated to the receiver. These public proving schemes can be done by either the signer or the receiver.

A. Signature Proving

Secret signature is a private transaction between a signer and a designated receiver, but if a dispute arises between them, they may want to prove the validity of secret signature to others. Signature proving is used to show that the secret signature generated by the signer is valid without revealing who is the receiver.

For signature proving signer or receiver just computes and reveals the agreed key

$$w = \hat{e}(pk, H(ID_R)^{k_1}) = \hat{e}(f, sk_{ID_R}).$$

Then anyone can verify that $\sigma = (u, f, v)$ is a correct signature of the signer S for a message m and some random-looking information w by

$$\hat{e}(v, g) \stackrel{?}{=} \hat{e}(pk, u \cdot H(ID_S)^{H'(m, ID_S, u, f, w)}).$$

B. Receiver Proving

Let's consider the following dispute scenarios;

- The receiver argues that he/she is not the receiver of the secret signature.
- The signer argues that he/she has not sent a secret signature to the receiver.

To resolve these disputes it is required to prove who is the receiver of the secret signature. Receiver proving can be used for this purpose by either the signer or the receiver. In receiver proving the agreed key w is revealed and it is proven that w is related with the designated receiver R in a special way.

In some case of dispute the prover (signer or receiver) may not want to reveal the fact that he/she is the prover. Thus we consider the following two cases of proof method.

- General Proof: In this proof method signer's proof and receiver's proof are distinguishable, thus the identity of the prover (signer or receiver) who proves the identity of the receiver is revealed.
- Anonymous Proof: As the name suggests, the identity of the prover who proves the identity of the receiver is not revealed. In this proof signer's proof and receiver's proof are indistinguishable. It is computationally more expensive than that of the general proof.

In the proposed IBSS scheme we use a modified version of key agreement scheme and a new proof technique in the receiver proving stage, where the receiver has to prove that the agreed key w is related with sk_{ID_R} and he has the secret key sk_{ID_R}.

General Proof. In this protocol, either the signer or the receiver, reveal the shared key w and prove its relation with the receiver using the proof of knowledge technique.

- **Signer's proof:**
  $\{ZKP(k_1) : (w = \hat{e}(pk, H(ID_R))^{k_1} \wedge f = g^{k_1})\}$
  The signer proves that w is related with $H(ID_R)$ using his knowledge of k_1. First, signer randomly chooses $t \in_R Z_q^*$. Then, it computes $w_1 = \hat{e}(pk, H(ID_R))^t$ and $f_1 = g^t$. He computes $c_1 = H'(w, w_1, f, f_1)$ and $s_1 = k_1 + c_1 t$, and sends (w_1, f_1, s_1) to the verifier.
  Then, verifier computes $c_1 = H'(w, w_1, f, f_1)$. Then he checks $\hat{e}(pk, H(ID_R))^{s_1} \stackrel{?}{=} w_1^{c_1} w$ and $g^{s_1} \stackrel{?}{=} f_1^{c_1} f$.
- **Receiver's proof:**
  $\{ZKP(sk_{ID_R}) : (w = \hat{e}(f, sk_{ID_R}))\}$
  The receiver with identity ID_R proves that w is related with $H(ID_R)$ using his knowledge of sk_{ID_R}. First, receiver randomly chooses $g_1 \in_R G_1$ and computes $w_2 = \hat{e}(g, g_1)$. He computes $c_2 = H'(w, w_2, f)$ and $s_2 = sk_{ID_R} g_1^{c_2}$, and sends (w_2, s_2) to the verifier.
  Then, verifier computes $c_2 = H'(w, w_2, f)$ and checks $\hat{e}(s_2, g) \stackrel{?}{=} \hat{e}(pk, H(ID_R)) w_2^{c_2}$.

This proof of knowledge of sk_{ID_R} is a variant of Schnorr type proof of knowledge. Note that this technique is also used in [12].

It is easy to see that the proofs initiated by the signer and the receiver are distinguishable, hence the identity of the prover is revealed.

Anonymous Proof. There might exist situations where we do not want to reveal the identity of the entity who proves the identity of receiver, perhaps, due to privacy or legal restrictions. In such cases, we cannot employ the general proof protocol presented above. Here we show how the signer or the receiver can prove the identity of receiver anonymously without revealing their identity.

The anonymous proof protocol can be initiated either by signer or by receiver. A prover reveals the shared key w and shows that it is related with the receiver R using the knowledge of secret information by

$$ZKP(k_1 \vee sk_{ID_R}) : \{w = \hat{e}(pk, H(ID_R))^{k_1} \wedge f = g^{k_1}\} \vee \{w = \hat{e}(f, sk_{ID_R})\}.$$

It is a OR combination of two zero-knowledge proofs of knowledge. It can be implemented as follows:

If the prover is the signer who has the secret k_1,

- It randomly chooses t, c_2, s_2 from Z_q^*;
- It computes $w_1 = \hat{e}(pk, H(ID_R))^t$, $f_1 = g^t$, and $w_2 = [\hat{e}(s_2, g)\hat{e}(pk, H(ID_R))^{-1}]^{c_2^{-1}}$;
- It computes $c_1 = H'(w, w_1, w_2, f, f_1) \oplus c_2$;
- Finally, compute $s_1 = k_1 + c_1 t$ and send $(w_1, w_2, f_1, c_1, c_2, s_1, s_2)$ to the verifier.

If the prover is the receiver who has the secret sk_{ID_R},

- It chooses $g_1 \in_R G_1$ and computes $w_2 = \hat{e}(g, g_1)$;
- It also chooses $c_1, s_1 \in_R Z_q^*$, computes $w_1 = [\hat{e}(pk, H(ID_R))^{s_1} w^{-1}]^{c_1^{-1}}$ and $f_1 = [g^{s_1} f^{-1}]^{c_1^{-1}}$;
- It computes $c_2 = H'(w, w_1, w_2, f, f_1) \oplus c_1$.
- Finally, compute $s_2 = sk_{ID_R} g_1^{c_2}$ and send $(w_1, w_2, f_1, c_1, c_2, s_1, s_2)$ to the verifier.

After receiving $(w_1, w_2, f_1, c_1, c_2, s_1, s_2)$, the verifier checks if the following four equations hold:

1) $c_1 \oplus c_2 \stackrel{?}{=} H'(w, w_1, w_2, f, f_1)$.
2) $\hat{e}(pk, H(ID_R))^{s_1} \stackrel{?}{=} w_1^{c_1} w$.
3) $g^{s_1} \stackrel{?}{=} f_1^{c_1} f$.
4) $\hat{e}(s_2, g) \stackrel{?}{=} \hat{e}(pk, H(ID_R))\ w_2^{c_2}$.

Although these two proofs by the signer and the receiver are computed differently, they are indistinguishable; any verifier is unable to distinguish whether the proof is provided by the signer or the receiver. If one of the party opens the secret signature anonymously, the other partner know that no one other than the partnering entity has opened the secret signature. However, the party is unable to convince others that the other partnering entity has opened the secret signature. From the public's perspective, the identity of prover remains anonymous.

V. Conclusion

In this paper we discussed the signature privacy issue and the secret signature scheme proposed by Lee *et al.* [11], and then presented an ID-based secret signature scheme. We first gave the security definition of IBSS in terms of unforgeability, invisibility, receiver designation, and public provability. To provide the public provability in secret signature scheme we modify Cha-Cheon's identity-based signature scheme [3] and identity-based key agreement scheme [14], and then combined them to implement ID-based secret signature scheme. We proved the security of the proposed IBSS scheme in the random oracle model under the standard CDH and DDH assumption. We also provided the signature proving and receiver proving protocols. Receiver proving is provided by using a modified proof of knowledge in pairing group.

References

[1] S. Araki, S. Uehara, and K. Imamura, "The Limited Verifier Signature and its Applications, IEICE Transactions,"The Institute of Electronics, Information and Communication Engineers Press, Japan, pp. 63–68, Volume E82, A(1), 1999.

[2] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing,"CRYPTO 2001, Springer, LNCS 2139, pp. 213–229, 2001.

[3] J. C. Cha and J. H. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Groups,"Public Key Cryptography 2003, LNCS 2567, pp. 18–30, 2003.

[4] D. Chaum, "Private Signature and Proof Systems,"United States Patent 5,493,614, 1996.

[5] L. Chen, Z. Cheng, and N.P. Smart, "Identity-based Key Agreement Protocols From Pairings,"ePrint, 2006/199.

[6] X. Chen, F. Zhang, K. Kim, "Limited Verifier Signature Scheme from Bilinear Pairing,"In ACNS 2004, LNCS 3089, pp. 135–148, Springer-Verlag, 2004.

[7] W. Diffie and M. E. Hellman, "New Directions in Cryptography,"In IEEE Transactions on Information Theory, volume IT-22(6), pp. 644–654, 1976.

[8] S. D. Galbraith and W. Mao, "Invisibility and Anonymity of Undeniable and Confimer Signatures,"In CT-RSA 2003, LNCS 2612, Springer-Verlag, pp. 80–97, 2003.

[9] S. Goldwasser, S. Micali, and R. L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks,"SIAM Journal on Computing, pp. 281–308, Society for Industrial and Applied Mathematics Press, Volume 17(2), 1988.

[10] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated Verifier Proofs and Their Applications,"EUROCRYPT 1996, LNCS 1070, pp. 321-331, Springer-Verlag, 1996.

[11] B. Lee, K. R. Choo, J. Yang, S. Yoo, "Secret Signatures: How to Achieve Business Privacy Efficiently?,"WISA 2007, LNCS 4867, pp. 30–47, Springer-Verlag, 2007.

[12] B. Libert, J.-J. Quisquater, "Identity Based Undeniable Signatures,"CT-RSA 2004, LNCS 2964, pp. 112–125, Springer-Verlag, 2004.

[13] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes,"CRYPTO'84, LNCS 196, pp. 47–53, Springer-Verlage, 1984.

[14] N. P. Smart, "An identity based authenticated key agreement protocol based on the Weil pairing,"Electronics Letters, 38, 630–632, 2002.

[15] Y. Zheng, "Digital Signcryption or How to Achieve Cost (Signature Encryption) $\ll$ Cost (Signature) + Cost (Encryption),"CRYPTO 1997, pp. 165–179, LNCS 1294, Springer-Verlag, 1997.