Efficient and Robust Secure Key Issuing Protocol in ID-based Cryptography

Byoungcheon Lee[1], Ed Dawson[2], SangJae Moon[3]

[1] Joongbu University,
101 Daehak-Ro, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea
[2] Information Security Research Centre,
Queensland University of Technology,
GPO Box 2434, Brisbane, QLD 4001, Australia
[3] Mobile Network Security Technology Research Center,
Kyungpook National University., Daegu, 702-701, Korea

Abstract. ID-based cryptosystems have many advantages over certificate-based cryptosystems in key distribution, but they also have an inherent drawback of key escrow problem, i.e., user's private key is known to the key generation center (KGC). Therefore secure key issuing (SKI) is an important issue in ID-based cryptography. In multiple authority approach [BF01,CHSS02], key generation function is distributed to multiple authorities, but multiple identifications by multiple authorities are required. Keeping key privacy using user-chosen secret information [Gen03,AP03] is a simple and efficient solution, but it loses the advantages of ID-based cryptosystems. Recently, Lee et al. [LBD04] proposed a secure key issuing protocol in which a private key is issued by a key generation center (KGC), and then its privacy is protected by multiple key privacy agents (KPAs). But it is not efficient, since it requires multiple serial computations and communications to issue a private key. Moreover Kwon [Kwo04] has shown that Lee et al.'s protocol is subject to impersonation attack and denial of service attack, since it has not used a secure signature scheme. In this paper we modify Lee et al.'s scheme and propose an efficient SKI protocol which provides efficiency and robustness by using t-out-of-n verifiable secret sharing (VSS) technique among multiple KPAs.

Keywords. Secure key issuing, ID-based cryptography, Bilinear pairing, Verifiable secret sharing, Robustness.

1 Introduction

ID-based cryptography. In traditional certificate-based public key cryptosystems, a user's public key is certified with a certificate issued by a certification authority (CA). Any participant who wants to use a public key must first verify the corresponding certificate to check the validity of the public key. When many CAs are involved between two users, trust relationship between those CAs also needs to be verified. Public key infrastructure (PKI) is an important infrastructure to manage the trust relationship between entities in a hierarchical manner. In certificate-based schemes key revocation is also a big issue. As a consequence, certificate-based public key cryptosystems require a large amount of storage and computing time to store, verify, and revoke certificates.

In 1984, Shamir [Sha84] proposed the ID-based cryptography which can greatly simplify key management problem. In ID-based cryptography an entity's public key is derived directly from its identity information, for example, name, e-mail address, or IP address of the user. The corresponding private key is generated for the user by a trusted third party called key generation center (KGC) and given to the user through a secure channel. Since Boneh and Franklin [BF01] have proposed secure ID-based encryption scheme (IBE) using bilinear pairing, ID-based cryptography has become more practical following many research results.

Compared with certificate-based cryptography, ID-based cryptography is advantageous in key management, since key distribution and key revocation are not required. A sender can send a secure message to a receiver just using the receiver's public identity information, even before the receiver obtains his private key from KGC. But an inherent problem of ID-based cryptography is the key escrow problem, i.e., KGC knows the user's private key. Therefore, KGC can decrypt any ciphertext and forge any entity's signature. It also requires a secure channel between users and KGC to deliver private keys. Because of these inherent problems ID-based cryptography is considered to be suitable only for small private network with lower security requirements, where KGC is fully trusted. Therefore providing a secure key issuing (SKI) mechanism in ID-based cryptography is an important issue to make it more applicable in the real world.

Secure key issuing. To tackle this problem, several proposals have been made using multiple authority approach [BF01,CHSS02] or using some user-chosen secret information [Gen03,AP03]. If the master key of a KGC is distributed to multiple authorities and a private key is computed in a threshold manner [BF01], key escrow problem of a single KGC can be prevented. However, multiple identifications for the same user by multiple authorities are required which is quite a burden in many cases. Correct identification is as important as correct key issuing. Generating a new private key by adding multiple private keys [CHSS02] is another approach, but in this scheme multiple identifications are also required and KGCs have no countermeasure against user's illegal usage. Gentry [Gen03] proposed a certificate-based encryption (CBE) scheme where secure key issuing was provided using some user-chosen secret information, but it became a certificate-based scheme losing the advantage of ID-based cryptography. [AP03] successfully removed the necessity of certificate (they named it certificateless public key cryptography) in a similar design using user-chosen secret information, but their scheme provides only implicit authentication of the public key. The public key generated by the user is not certified in any way. Thus any participant who wants to use the public key cannot be convinced whether the public key indeed belongs to the user.

Recently, Lee et al. [LBD04] proposed a unique secure key issuing protocol in which a private key is issued by a single key generation center (KGC) and then its privacy is protected by multiple key privacy agents (KPAs). This proposal reduced the identification cost, since it requires a single identification by KGC. But it is not efficient, since it requires serial computations and communications by multiple KPAs to issue a private key. It is not robust since a single failure or unavailability among multiple authorities will cause a failure in key issuing. Moreover Kwon [Kwo04] has shown that Lee et al.'s protocol is subject to impersonation attack and denial of service attack. This weakness comes from the fact that Lee et al.'s scheme has not used a secure signature scheme and each KPAs cannot verify the validity of previous result in the serial execution chain.

Real world scenarios. In the real world a specific authority is generally given to a single authority. For example, a driver's license is issued by a single authority, although there can be many regional offices for efficiency. Correct identification of user is as important as correct issuing. When the identification is critical, the authority can require physical presence of the user. The single authority approach is easy to implement and acceptable in ordinary certificate-based schemes, but it suffers from key escrow problem when used in ID-based cryptography. It seems to be inevitable to introduce multiple authorities to avoid key escrow problem in ID-based cryptography.

In this paper we consider the *single authority-multiple observer (SAMO)* model, i.e., a single entity has the authority, but other multiple entities observe (or supervise) the single authority. This model seems to be quite reasonable since we can find several good examples in the real world.

First, we can consider the example of non-governmental organizations (NGOs) or ombudsman who are organized by themselves (not by the government) and are trying to observe (or supervise) whether the government does anything illegal. They do not have any authority given by the government, but their roles of supervising the government are generally accepted by the people if their activities are sound enough. If the government has a super-power like a big brother, the role of NGOs are very important. Although they do not have a legally approved authority like governmental organizations, they can provide service to prevent government's misbehavior.

Another example can be found in elections. In a political election there is a single election administrator who organizes and manages the election processes, but major political parties send their observers to the voting office to detect any illegal activity; illegal voters, double voting, threatening, miscounting, etc. Since each political party has different interest in the election, it is hard to assume that all observers collude together. The supervision of voting and counting processes by the observers from political parties are generally accepted in many countries. If there are some possibilities of misbehavior by the administrator, the role of observers becomes very important.

We consider that if there are multiple entities like NGOs or observers, KPAs in this paper, who provide privacy service for user's private key in ID-based cryptosystems, then there is a way that a single KGC can issue the ID-based private key in a secure manner. Note that KPAs are not authorities but voluntary observers, thus the role of KPAs needs to be robust, i.e., partial failure among multiple KPAs should not cause any problem.

Our contribution. In this paper we modify Lee et al.'s scheme and propose an efficient SKI protocol which provides efficiency and robustness by using t-out-of-n verifiable secret sharing technique among multiple KPAs. In this proposal computations and communications by multiple KPAs can be done in parallel and public manner, which significantly improves efficiency. Partial failure of KPAs is allowed by using the secret sharing technique.

The rest of the paper is organized as follows. Background concepts such as bilinear pairing, ID-based cryptography, short signature, batch verification, and secret sharing in pairing cryptography are briefly reviewed in Section 2. Proposed efficient key issuing protocol and key escrow protocol are described in Section 3 and 4, respectively. We analyze the proposed scheme in Section 5 and finally conclude in Section 6.

2 Cryptographic Primitives

In this Section we briefly review the basic concepts of bilinear pairing, ID-based cryptography, short signature, batch verification, and secret sharing in pairing cryptography, and introduce the basic notation used in this paper.

2.1 Bilinear Pairing

Let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order. Let P denote a generator of G_1. The discrete logarithm problem (DLP) in these groups is believed to be hard. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. Bilinear: $e(aQ_1, bQ_2) = e(Q_1, Q_2)^{ab}$, where $Q_1, Q_2 \in G_1$ and $a, b \in Z_q^*$.
2. Non-degenerate: $e(P, P) \neq 1$ and therefore it is a generator of G_2.
3. Computable: There is an efficient algorithm to compute $e(Q_1, Q_2)$ for all $Q_1, Q_2 \in G_1$.

We write G_1 with an additive notation and G_2 with a multiplicative notation, since in general implementation G_1 will be the group of points on an elliptic curve and G_2 will denote a multiplicative subgroup of a finite field. Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. We refer to [BF01,BKLS02] for a more comprehensive description on how these groups, pairings and other parameters should be selected for efficiency and security.

Now we describe some mathematical problems.

- Discrete Logarithm Problem (DLP): Given two group elements P and Q in G_1, find an integer n, such that $Q = nP$ whenever such an integer exists.

- Computational Diffie-Hellman Problem (CDHP): Given $\langle P, aP, bP \rangle$ for any $a, b \in Z_q^*$, compute abP.
- Decisional Diffie-Hellman Problem (DDHP): Given $\langle P, aP, bP, cP \rangle$ for any $a, b, c \in Z_q^*$, decide whether $c \equiv ab \bmod q$.
- Bilinear Diffie-Hellman Problem (BDHP): Given $\langle P, aP, bP, cP \rangle$ for any $a, b, c \in Z_q^*$, compute $e(P, P)^{abc} \in G_2$.
- Gap Diffie-Hellman Problem (GDHP): A class of problems where DDHP is easy while CDHP is hard.

In this paper we consider the GDHP group where the DDHP is easy but the CDHP is hard. Such groups can be found on supersingular elliptic curves or hyperelliptic curves over a finite field. The bilinear pairing described above is a good example.

2.2 ID-based Cryptography

Using the bilinear pairing, ID-based encryption scheme can be designed easily. In ID-based cryptography, there is a trusted authority called the key generation center (KGC) who has a master key s_0 and issues private keys for users. Boneh and Franklin's "BasicIdent"scheme [BF01] is given by the following four stages.

Setup: KGC specifies two groups G_1 and G_2 and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ between them. It also specifies three hash functions.

- $H_1 : \{0, 1\}^* \rightarrow G_1$ (extract point from ID).
- $H_2 : G_2 \rightarrow \{0, 1\}^l$, where l is the length of a plaintext message (hash to the message space).
- $H_3 : \{0, 1\}^* \rightarrow Z_q^*$ (hash to the finite field, which will be used in the proposed key issuing protocol).

KGC picks a master key $s_0 \in Z_q^*$ at random and computes his public key $P_0 = s_0P$. KGC publishes description of the groups G_1, G_2, bilinear map e, hash functions H_1, H_2, H_3, and his public key P_0.

Extract: Let Alice be a sender and Bob be a receiver. Bob requests KGC to issue a private key for his $ID \in \{0, 1\}^*$. For given Bob's identity ID, KGC computes Bob's public key as $Q_{ID} = H_1(ID)$ and the corresponding private key as $D_{ID} = s_0Q_{ID}$. Note that D_{ID} is a short signature [BLS01] of the KGC on the message ID. Then he sends D_{ID} to Bob through a secure channel. Bob can check the validity of his private key by $e(D_{ID}, P) \overset{?}{=} e(Q_{ID}, P_0)$.

Encrypt: To encrypt a message $m \in \{0, 1\}^l$ with the public key of the receiver Bob, Alice first computes Bob's public key by $Q_{ID} = H_1(ID)$. Then she picks a random number $r \in Z_q^*$ and computes $U = rP$ and $V = m \oplus H_2(e(Q_{ID}, P_0)^r)$. Then the ciphertext $C = (U, V)$ is sent to Bob.

Decrypt: The receiver Bob can decrypt the ciphertext $C = (U, V)$ using his private key D_{ID} by $V \oplus H_2(e(D_{ID}, U)) = m$. The decryption works because of the bilinear property of the map e,

$$e(D_{ID}, U) = e(s_0Q_{ID}, rP) = e(Q_{ID}, P_0)^r.$$

2.3 Short Signature

Assume that a signer has a private key s_0 and corresponding public key $P_0 = s_0 P$. A short signature on message m is $Sig = s_0 H_1(m)$. A verifier can check the validity of Sig by

$$e(Sig, P) \overset{?}{=} e(H_1(m), P_0).$$

This signature scheme is secure against existential forgery under a chosen-message attacks in the random oracle model [BLS01].

2.4 Verifiable Secret Sharing in Pairing Cryptography

Let $s_K \in Z_q^*$ be the secret key shared by n KPAs and $P_K = s_K P$ be the corresponding public key. The secret information s_K can be distributed among n KPAs in t-out-of-n fashion using the verifiable secret sharing (VSS) scheme [Ped91,GJK99]. It is straightforward to extend VSS scheme to pairing cryptosystems. Assume that following the VSS protocol KPA_i finally has a secret share s_i and publishes the corresponding public share $P_i = s_i P$. Then the public key of KPAs can be computed by

$$P_K = \sum_{i \in \Lambda} \lambda_{i,\Lambda} P_i,$$

where $\lambda_{i,\Lambda} = \prod_{l \in \Lambda \setminus \{i\}} \frac{l}{l-i}$ is the appropriate Lagrange coefficient and Λ is a subset of valid t public shares.

2.5 Batch Verification

Assume that n KPAs have secret shares s_i and public shares $P_i = s_i P$ for $i = 1, \ldots, n$. If they have computed scalar multiplications for some message U like $U_i = s_i U$, its correctness can be checked immediately by $e(U_i, P) \overset{?}{=} e(U, P_i)$ (Note that if discrete log based cryptography is used for secret sharing scheme, zero-knowledge proofs are needed to prove the correctness of share computations). For the verification of n scalar multiplications by n KPAs, $2n$ pairing computations are required. But if we use the batch verification technique [BGR98], we can reduce the number of pairing computations drastically.

A verifier chooses small numbers $t_1, \ldots, t_n$ (for example, $|t_i| \sim 20$ bits) and checks

$$e(t_1 U_1 + t_2 U_2 + \cdots + t_n U_n, P) \overset{?}{=} e(U, t_1 P_1 + t_2 P_2 + \cdots + t_n P_n),$$

which requires just two pairing computations and $2n$ small number scalar multiplications. The probability that an incorrect set of U_is can pass this verification is $2^{-20} \sim 10^{-6}$.

3 Proposed Efficient Secure Key Issuing Protocol

In this paper we propose a new efficient and robust secure key issuing protocol using the SAMO model, which preserves the advantages of ID-based cryptography. We assume the existence of a single key generation center (KGC) who issues a partial private key and multiple key privacy agents (KPAs) who provide key privacy service. The entities participating in the secure key issuing protocol and their roles are as follows.

- KGC : A single KGC is a trusted authority who checks user's identification and issues a blinded partial private key to the user.
- n KPAs: Multiple KPAs are voluntary service agents who share the secret key s_K using the t-out-of-n VSS scheme. We assume that at least t KPAs will remain honest and available. Partial failure or unavailability of up to $n-t$ KPAs is allowed. They provide key privacy service to user's private key by issuing their signature in a blinded manner for the blinded partial private key generated by KGC.
- User: Through an interactive protocol with KGC and n KPAs, user finally gets a private key for ID-based cryptosystems in a secure way.

Under a court order KGC and KPAs can cooperate to provide a key escrow service for a specific message.

In this proposal a user requests key issuing to a single KGC with his identity and blinding factor. Then, after checking the identification of the user, KGC issues a partial private key to the user in a blinded manner. After receiving the blinded partial private key, the user requests key privacy service to n KPAs in a parallel manner, then KPAs returns signed messages in a blinded manner. Finally, the user unblinds them and retrieves the real private key using the t-out-of-n VSS technique.

The proposed secure key issuing protocol consists of the following 5 stages; system setup, system public key setup, partial key issuing, key securing, and key retrieving stages.

Stage 1. System setup (by KGC)

As shown in Section 2, KGC specifies two groups G_1 and G_2 and the bilinear map $e : G_1 \times G_1 \rightarrow G_2$ between them, and three hash functions H_1, H_2, H_3. He publishes description of the groups G_1, G_2, the bilinear map e, and hash functions H_1, H_2, H_3.

Stage 2. System public key setup (by KGC and KPAs)

KGC picks his secret key $s_0 \in Z_q^*$ at random and publishes his public key $P_0 = s_0 P$. As shown in Section 2, n KPAs share the KPA's secret key s_K in t-out-of-n fashion using the VSS scheme such that KPA_i has a secret share s_i and publishes a public share $P_i = s_i P$. KPA's public key $P_K = s_K P$ is computed and published. Then KGC computes the system public key $Y = s_0 P_K = s_0 s_K P$ and publishes it, which will be used as a system public key for ID-based cryptosystems. Anyone can verify the validity of Y by $e(Y, P) \overset{?}{=} e(P_K, P_0)$.

Stage 3. Partial key issuing (by KGC and user)

A user with identity ID chooses a random secret x and computes $X = xP$. He computes a short signature for message (ID, X, KGC) with a secret x as $Sig_x(ID, X, KGC) = xH_1(ID, X, KGC)$. Then he identifies himself to KGC using proper online or offline procedure (predetermined depending on applications) and requests to issue a partial private key by sending $\langle ID, X, Sig_x(ID, X, KGC)\rangle$. Here $Sig_x(ID, X, KGC)$ represents the proof of possession of the secret x corresponding to X. Then KGC issues a blinded partial private key as follows;

1. checks the identification of the user using proper online or offline procedure (predetermined depending on applications).
2. checks the proof of possession of x by
$$e(Sig_x(ID, X, KGC), P) \overset{?}{=} e(H_1(ID, X, KGC), X).$$
3. computes the public key of the user as
$$Q_{ID} = H_1(ID, KGC, KPA_1, \ldots, KPA_n).$$
4. computes a blinded partial private key as
$$Q_0' = H_3(P_0, X, s_0X)s_0Q_{ID}.$$
5. computes a signature on (Q_0', ID, X) as
$$Sig_0(Q_0', ID, X) = s_0H_1(Q_0', ID, X).$$
6. sends $\langle Q_0', Sig_0(Q_0', ID, X)\rangle$ to the user.

Here $H_3(P_0, X, s_0X)$ is a kind of secure channel between user and KGC. Using the same Diffie-Hellman key $s_0X = xP_0$, KGC and user can communicate securely. User unblinds it to compute

$$Q_0 = \frac{Q_0'}{H_3(P_0, X, xP_0)} = s_0Q_{ID},$$

and checks its validity by

$$e(Q_0, P) \overset{?}{=} e(Q_{ID}, P_0).$$

Stage 4. Key securing (by user and KPAs)

If Q_0 turns out to be valid, user computes an approval message as follows signing with x.

$$Sig_x(Q_0', ID, X) = xH_1(Q_0', ID, X).$$

User sends $\langle ID, X, Q_0', Sig_0(Q_0', ID, X), Sig_x(Q_0', ID, X)\rangle$ to n KPAs and requests key privacy service. Then each KPA_i

1. checks KGC's signature as

$$e(Sig_0(Q'_0, ID, X), P) \overset{?}{=} e(H_1(Q'_0, ID, X), P_0).$$

2. checks user's signature as

$$e(Sig_x(Q'_0, ID, X), P) \overset{?}{=} e(H_1(Q'_0, ID, X), X).$$

3. computes $Q'_i = H_3(P_i, X, s_iX))s_iQ'_0$ and

$$Sig_i(Q'_i, Q'_0, ID, X) = s_iH_1(Q'_i, Q'_0, ID, X).$$

4. sends $\langle Q'_i, Sig_i(Q'_i, Q'_0, ID, X)\rangle$ to the user.

Here $H_3(P_i, X, s_iX)$ is a secure channel between user and KPA_i. Using the same Diffie-Hellman key $s_iX = xP_i$, KPA_i and user can communicate securely.

Stage 5. Key retrieving (by user)

After receiving $\langle Q'_i, Sig_i(Q'_i, Q'_0, ID, X)\rangle$ from each KPA_i, user

1. checks the validity of each signature by

$$e(Sig_i(Q'_i, Q'_0, ID, X), P) \overset{?}{=} e(H_1(Q'_i, Q'_0, ID, X), P_i).$$

2. unblinds each Q'_i and computes

$$Q_i = \frac{Q'_i}{H_3(P_0, X, xP_0)H_3(P_i, X, xP_i)} = s_iQ_0 = s_is_0Q_{ID}.$$

3. checks the validity of each Q_i by

$$e(Q_i, P) \overset{?}{=} e(Q_0, P_i).$$

4. retrieves his private key by computing

$$D_{ID} = \sum_{i \in \Lambda} \lambda_{i,\Lambda} Q_i = s_K s_0 Q_{ID},$$

where $\lambda_{i,\Lambda} = \prod_{l \in \Lambda \setminus \{i\}} \frac{l}{l-i}$ is the appropriate Lagrange coefficient and Λ is a subset of t valid Q_is.

5. verifies the correctness of his private key by $e(D_{ID}, P) \overset{?}{=} e(Q_{ID}, Y)$.

Assuming the honesty of at least t KPAs, the privacy of user's private key D_{ID} is attained. Only the legitimate user who knows the random secret x can unblind the protocol messages to recover the private key.

In stages 3 and 4, $Sig_0(Q'_0, ID, X)$ and $Sig_i(Q'_i, Q'_0, ID, X)$ represent that Q'_0 and Q'_i are generated by KGC and KPA_i, respectively. $Sig_x(Q'_0, ID, X)$ represents that user approves Q'_0. KPA_i is convinced from $Sig_0(Q'_0, ID, X)$ and $Sig_x(Q'_0, ID, X)$ that Q'_0 was issued by KGC and approved by user. Only when they are valid, KPA_i provides his privacy service by generating Q'_i. User or any observer is convinced from $Sig_i(Q'_i, Q'_0, ID, X)$ that Q'_i was issued by KPA_i. Thus, these signatures have the role of binding the information (ID, X, Q'_0, Q'_i) and making the protocol messages authentic.

Since the interactive protocol messages in stages 3 and 4 are blinded using X, all messages can be published, for example, on a bulletin board. User requests key issuing by publishing $\langle ID, X, Sig_x(ID, X, KGC)\rangle$, KGC issues blinded partial private key by publishing $\langle Q'_0, Sig_0(Q'_0, ID, X)\rangle$, user approves Q'_0 by publishing $Sig_x(Q'_0, ID, X)$, and KPA_i provides key privacy service by publishing $\langle Q'_i, Sig_i(Q'_i, Q'_0, ID, X)\rangle$. Then user checks the correctness of each job and retrieves his private key from correct messages. If each entity provides his service only in a public way as shown above (public job model), the possibility of illegal activities by potential attackers is reduced and each entity has to be careful to keep his reputation and business.

To check the validity of n Q_is independently, $2n$ pairing computations are required. But, if the batch verification technique is used, the number of pairing computations can be reduced a lot. User chooses small numbers $t_1, \ldots, t_n$ (for example, $|t_i| \sim 20$ bits) and checks

$$e(t_1Q_1 + t_2Q_2 + \cdots + t_nQ_n, P) \overset{?}{=} e(Q_0, t_1P_1 + t_2P_2 + \cdots + t_nP_n),$$

which requires just two pairing computations if the $2n$ small number scalar multiplications are not considered.

The private key D_{ID} retrieved in this way is a real ID-based private key corresponding to the public key Q_{ID} when $Y = s_0s_KP$ is used as the system public key. Therefore this key pair can be used for any ID-based cryptosystems, such as encryptions [BF01], signatures [CC03], etc. The proposed secure key issuing protocol overcomes the key escrow problem of ID-based cryptography, thus it can be applied to more complex applications with strong security requirements.

4 Key Escrow Protocol

The proposed protocol supports key escrow per message under a court order. Assume that a ciphertext

$$C = (U, V) = (rP, m \oplus H_2(e(Q_{ID}, Y)^r))$$

is given which is an encryption of a message m with the public key Q_{ID}. Then user's decryption will be given by

$$V \oplus H_2(e(D_{ID}, U)) = m.$$

Under a court order, KGC and n KPAs can cooperate to decrypt the ciphertext C to recover the plaintext m without disclosing D_{ID}. First, each KPA_i computes $U_i = s_iU$. Its correctness can be verified immediately by $e(U_i, P) \overset{?}{=} e(U, P_i)$. Let Λ be a subset of t valid U_is. Then KGC computes

$$U_K = \sum_{i \in \Lambda} \lambda_{i,\Lambda} U_i = s_KU,$$

where $\lambda_{i,\Lambda} = \prod_{l \in \Lambda \setminus \{i\}} \frac{l}{l-i}$ is the appropriate Lagrange coefficient. Finally KGC computes $U' = s_0 U_K = s_0 s_K U$. Then the plaintext message can be recovered by

$$V \oplus H_2(e(Q_{ID}, U')) = m.$$

To check the validity of n U_is independently, $2n$ pairing computations are required. But, if the batch verification technique is used, the number of pairing computations can be reduced a lot. The user chooses small numbers $t_1, \ldots, t_n$ (for example, $|t_i| \sim 20$ bits) and checks

$$e(t_1U_1 + t_2U_2 + \cdots + t_nU_n, P) \overset{?}{=} e(U, t_1P_1 + t_2P_2 + \cdots + t_nP_n),$$

which requires just two pairing computations.

5 Analysis

The proposed scheme is a secure key issuing protocol. Since KPA's private key s_K was shared among n KPAs using a t-out-of-n VSS scheme, the privacy of user's private key D_{ID} is attained if we assume the honesty of at least t KPAs. KGC, who is the most advantageous entity in the protocol, cannot get any useful information without help of more than t KPAs. Any attacker who tries to get any useful information from the protocol messages will not be successful, since every protocol messages are blinded with user-chosen secret using the non-interactive Diffie-Hellman key agreement technique. Only the legitimate user who knows the random secret x can unblind the protocol messages to recover the private key. It also has robustness. Any partial failure or unavailability up to $n-t$ KPAs is allowed.

[Kwo04] pointed out that in Lee et al.'s scheme [LBD04] KGC can retrieve user's private key by launching an attack using a new blinding factor Z chosen by himself instead of user-chosen blinding factor X. This attack is possible since [LBD04] has not used a secure signature scheme in the protocol and each entity cannot verify the validity of previous result in the serial execution by KGC and n KPAs. But in the proposed scheme a secure signature scheme [BLS01] was used and each entity can verify the validity of previous result. KPA_i will provide privacy service only if KGC's signature and user's approval message are all valid. Thus [Kwo04]'s attack is not possible in this protocol.

Initially user was not certified in any way, thus user's signing with x is just a proof of possession of the secret corresponding to X. If KGC prepares the protocol messages of the stage 3 by himself with the name of user (but without any interaction or permission of the user) and requests key privacy service to KPAs, they will provide key privacy service and KGC will be able to get a valid key of the user. There is no way to prevent this attack, which is the same case in traditional certificate-based schemes. If a certification authority is not trusted, he can issue a certificate by himself with any key and identity. Therefore we assume that KGC is a trusted authority (like a certification authority in certificate-based cryptography) who cannot do this kind of attack. Since all blinded protocol messages are published in the proposed scheme, the correctness of protocol is publicly verifiable. In this public job model KGC cannot do this kind of illegal activity in a public way. He has to be careful to keep his reputation and business.

In [LBD04], each KPA_i provides his privacy service in a serial way by just checking the signature of previous entity KPA_{i-1}. If some of the previous entities have sent incorrect results with correct signatures (each KPA_i cannot verify the correctness of previous result), the final result will be incorrect. In the proposed scheme user can verify the correctness of every protocol messages from KGC and KPAs (he can unblind messages by using x), and KPAs provide their privacy service after checking KGC's signature and user's approval messages.

We compare the efficiency and features of proposed scheme with [BF01], [CHSS02], and [LBD04] in Table 1.

- Compared with [BF01], and [CHSS02], the proposed scheme and [LBD04] reduce the identification cost from n to 1. In many cases in the real world key issuing authority can require physical presence of the user, so this improvement is very important.
- Computations and communications between user and KPAs are parallel in the proposed scheme, while it was serial in [LBD04].
- The proposed scheme is robust in the sense that partial failure among multiple KPAs are allowed, while [LBD04] is not robust. In the proposed scheme the corporation of t KPAs is enough to provide key privacy service and partial failure of up to $n-t$ KPAs is allowed.

One more improvement is that the proposed scheme does not use pairing computation to construct secure channels, while [LBD04] has used pairing computation.

Table 1. Comparison of efficiency.

	[BF01]	[CHSS02]	[LBD04]	Proposed
Identification cost	n	n	1	1
Communication model	Parallel	Parallel	Serial	Parallel
Robustness	Yes	No	No	Yes

6 Conclusion

In this paper we modify Lee et al.'s SKI protocol [LBD04] and propose an efficient and robust SKI protocol using t-out-of-n VSS technique. We used the single authority-multiple observer (SAMO) model in which a single KGC issues a partial private key in a blinded manner and then multiple KPAs provide key privacy service in a parallel manner. This approach seems to be quite reasonable and useful in many applications, since we can find several good examples in the real world. Applying this method real world authorities can be used in more distributed way.

The proposed SKI scheme guarantees key privacy if at least t KPAs remain honest. It is robust in the sense that partial failure of up to $n - t$ KPAs is allowed. Using a simple blinding technique with a non-interactive Diffie-Hellman key agreement technique, a secure channel is provided. Only the legitimate user who has the random secret x corresponding to X can recover the private key by unblinding the protocol messages. Since the protocol messages are blinded with X, all messages can be published. Using this public job model, the correctness of every job can be publicly verifiable and the possibility of illegal activities by potential attackers is reduced.

The proposed SKI scheme is advantageous over previous schemes since it requires single identification by KGC, parallel computations and communications by multiple KPAs are possible, and partial failure of up to $n-t$ KPAs is allowed.

The generated key using the proposed key issuing protocol is a real ID-based private key, therefore it can be used for any ID-based cryptosystems such as encryptions, signatures, and key agreements. Using the proposed SKI scheme ID-based cryptography can be more practical.

Acknowledgements

This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA.

References

[AP03] S. Al-Riyami, K. Paterson, "Certificateless public key cryptography", Advances in Cryptology – Asiacrypt'2003, LNCS 2894, Springer-Verlag, pp. 452–473, 2003.

[BF01] D. Boneh, and M. Franklin, "Identity-based encryption from the Weil pairing", Advances in Cryptology – Crypto'2001, LNCS 2139, Springer-Verlag, pp. 213–229, 2001.

[BGR98] M. Bellare, J. Garay and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures", *Advances in Cryptology – Eurocrypt'98*, LNCS 1403, Springer-Verlag, pp. 236–250, 1998.

[BKLS02] P. Barreto, H. Kim, B. Lynn, M. Scott, "Efficient algorithms for pairing-based cryptosystems", Advances in Cryptology – Crypto'2002, LNCS 2442, Springer-Verlag, pp. 354–368, 2002.

[BLS01] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing", Advances in Cryptology – Asiacrypt'2001, LNCS 2248, Springer-Verlag, pp. 514–532, 2002.

[CC03] J. Cha and J. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Groups", *Practice and Theory in Public Key Cryptography – PKC'2003*, LNCS 2567, Springer-Verlag, pp. 18–30, 2003.

[CHSS02] L. Chen, K. Harrison, N. P. Smart, D. Soldera, "Applications of multiple trust authorities in pairing based cryptosystems", InfraSec 2002, LNCS 2437, Springer-Verlag, pp. 260–275, 2002.

[Gen03] C. Gentry, "Certificate-based encryption and the certificate revocation problem", Advances in Cryptology - EUROCRPYT 2003, LNCS 2656, Springer-Verlag, pp. 272 – 293, 2003.

[GJK99] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems", Advances in Cryptology Eurocrypt '99, Lecture Notes in Computer Science, Vol. 1592, Springer-Verlag, pp. 295–310, 1999.

[Kwo04] S. Kwon, "Cryptanalysis for Secure Key Issuing in ID-based Cryptography and Improvement", Manuscript. 2004.

[LBD04] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang and S. Yoo, "Secure Key Issuing in ID-Based Cryptography", In ACSW Frontiers 2004 - Second Australasian Information Security Workshop (AISW 2004), volume 26 of Australian Computer Science Communications, pages 66–74. Australian Computer Society, January 2004.

[Pat02] K. Paterson, "Cryptography from pairings: a snapshot of current research", Information Security Technical Report, Vol. 7(3), pp. 41-54, 2002.

[Ped91] T. Pedersen, "A threshold cryptosystem without a trusted party", *Advances in Cryptology – Eurocrypt'91*, LNCS 547, Springer-Verlag, pp. 522–526, 1991.

[Sha84] A. Shamir, "Identity based cryptosystems and signature schemes", Advances in Cryptology - Crypto'84, LNCS 196, Springer-Verlag, pp. 47–53, 1984.