Secure Key Issuing in ID-based Cryptography

Byoungcheon Lee[1,2] **Colin Boyd**[1] **Ed Dawson**[1] **Kwangjo Kim**[3]
Jeongmo Yang[2] **Seungjae Yoo**[2]

[1]Information Security Research Centre,
Queensland University of Technology,
GPO Box 2434, Brisbane, QLD 4001, Australia,
Email: {b6.lee,c.boyd,e.dawson}@qut.edu.au

[2]Joongbu University,
101 Daehak-Ro, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea,
Email: {sultan,jmyang,sjyoo}@joongbu.ac.kr

[3]Information and Communications University,
58-4, Hwaam-dong, Yusong-gu, Daejeon, 305-732, Korea,
Email: kkj@icu.ac.kr

Abstract

ID-based cryptosystems have many advantages over PKI based cryptosystems in key distribution, but they also have an inherent drawback of key escrow problem, i.e. users' private keys are known to the key generation center (KGC). Therefore secure key issuing (SKI) is an important issue in ID-based cryptography. In multiple authority approach (Boneh & Franklin 2001, Chen et al. 2002), key generation function is distributed to multiple authorities. Keeping key privacy using user-chosen secret information (Gentry 2003, Al-Riyami & Paterson 2003) is a simple and efficient solution, but it loses the advantages of ID-based cryptosystems.

In this paper we propose a new 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 authorities (KPAs). In this protocol we provide a secure channel by using simple blinding technique in pairing-based cryptography. Only a legitimate user who has the secret blinding parameter can retrieve his private key from the protocol.

Keywords: Bilinear pairing, ID-based cryptography, Secure key issuing (SKI), Key generation center (KGC), Key privacy authority (KPA), Blinding.

1 Introduction

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 relationships between those CAs also need 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, which requires a large amount of storage and computing. As a consequence, certificate-based public key cryptosystems require a large amount of storage and computing time to store, verify, and revoke certificates.

(Shamir 1984) proposed the ID-based cryptography which can greatly simplify the 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.

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 identity information, even before the receiver obtains his private key from the KGC. But an inherent problem of ID-based cryptography is the key escrow problem, i.e., user's private key is known to the KGC. Therefore, the KGC can decrypt any ciphertext and forge signature for any message, so there is no user privacy and authenticity in the system. It also requires a secure channel between users and the 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. Therefore providing a secure key issuing mechanism in ID-based cryptography is an important issue to make the ID-based cryptography more applicable to the real world.

To tackle this problem, several proposals have been made using multiple authority approach (Boneh & Franklin 2001, Chen et al. 2002) or using some user-chosen secret information (Gentry 2003, Al-Riyami & Paterson 2003). If the master key of a KGC is distributed to multiple authorities and a private key is computed in a threshold manner (Boneh & Franklin 2001), key escrow problem of a single KGC can be prevented. However, in many applications multiple identifications of user by multiple authorities is quite a burden. Generating a new private key by adding multiple private keys (Chen et al. 2002) is another approach, but in this scheme KGCs have no countermeasure against user's illegal usage. Gentry (Gentry 2003) proposed a certificate-based encryption 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. (Al-Riyami & Paterson 2003) 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 securely generated by the user is not certified in any way. Thus any participant using the public key cannot be convinced of whether the public key indeed belongs to the user. In Section 3, we will review these schemes and discuss their properties in more detail.

In this paper we propose a new 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 authorities (KPAs). In this protocol we provide a secure channel by using a simple blinding technique in pairing-based cryptography. Only a legitimate user who has the secure blinding parameter can retrieve his private key from the protocol. In the proposed scheme user-chosen secret information is used, but it is used only for blinding purpose. The proposed secure key issuing protocol issues a real ID-based private key, thus it can be used with any ID-based cryptosystems preserving the advantage of ID-based cryptography.

The rest of the paper is organized as follows. Background concepts on bilinear pairing and ID-based cryptography are briefly reviewed in Section 2 and related works on secure key issuing are described in Section 3. Our key issuing model is introduced in Section 4. We describe the proposed secure key issuing protocol and key escrow protocol in Section 5 with security analysis. Finally, we conclude in Section 6.

2 Background Concepts

In this Section we briefly review the basic concepts on bilinear pairing and ID-based cryptography, while introducing notations 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 (Boneh & Franklin 2001, Barreto et al. 2002) 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): For any $a, b \in Z_q^*$, given $\langle P, aP, bP \rangle$, compute abP.
- Decisional Diffie-Hellman Problem (DDHP): For any $a, b, c \in Z_q^*$, given $\langle P, aP, bP, cP \rangle$, decide whether $c \equiv ab \bmod q$.
- Bilinear Diffie-Hellman Problem (BDHP): For any $a, b, c \in Z_q^*$, given $\langle P, aP, bP, cP \rangle$, 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 and the GDHP, ID-based encryption scheme can be designed very easily. In ID-based cryptography, there is a trusted authority called the key generation center (KGC) who has a master key s and issues private keys for users. Boneh and Franklin's "BasicIdent"scheme (Boneh & Franklin 2001) 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).
- $H_3 : G_2 \rightarrow Z_q^*$ (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 $P_0 = s_0 P$. KGC publishes description of the groups G_1, G_2, the bilinear map e, the 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 requires a private key for his $ID \in \{0, 1\}^*$ to KGC. For given Bob's identity ID, the KGC computes Bob's public key as $Q_{ID} = H_1(ID)$ and the corresponding private key as $D_{ID} = s_0 Q_{ID}$. Note that D_{ID} is a short signature (Boneh, Lynn & Shacham 2002) 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,

$$e(D_{ID}, U) = e(s_0 Q_{ID}, rP) = e(Q_{ID}, P_0)^r.$$

3 Related Works on Secure Key Issuing

To provide a secure key issuing there have been two approaches; using multiple authorities (Boneh & Franklin 2001, Chen et al. 2002) and using some user-chosen secret information (Gentry 2003, Al-Riyami & Paterson 2003). In this Section we review these works briefly and discuss their properties.

3.1 Threshold Key Issuing

Boneh and Franklin considered the distributed key generation in their original proposal of identity-based encryption (Boneh & Franklin 2001) to protect the secrecy of the master key s_0, not to protect the privacy of the private keys of users. Assume that there are n KGCs instead of a single KGC. The master key s_0 can easily be distributed in a t-out-of-n fashion by giving each of the n KGCs one share s_i of a Shamir secret sharing of s_0. When generating a private key, each of the t chosen KGCs respond with $Q_i = s_i Q_{ID}$. Then the user can construct $D_{ID} = \sum \lambda_i Q_i$, where the λ_i's are the appropriate Lagrange coefficients. This scheme can be made robust against dishonest KGCs without using zero-knowledge proofs, as shown in (Boneh & Franklin 2001).

If the master key of a KGC is distributed to multiple KGCs and the private key of a user is computed in a threshold manner as shown above, key escrow problem of a single KGC can be prevented. This approach is mathematically beautiful. However, in this approach, multiple KGCs are assumed to have the same role, so they have to check user's identity independently, which is quite a burden. Correct identification of user is as important as correct signing. As a tradeoff, a single KGC can check user's identity and other $n-1$ KGCs just sign ID, it is not an ideal distributed key generation. In this approach a secure channel is still required, although it can be provided easily using the simple blinding technique adapted in the proposed scheme.

3.2 Addition of Multiple Keys

To avoid the built-in key escrow of ID-based cryptography, (Chen et al. 2002), (Paterson 2002) and (Hess 2002) assumed multiple KGCs, each providing ID-based key issuing independently in the same group, and proposed for the user to generate a new private key securely by adding multiple independent private keys issued by multiple KGCs.

Assume that there are n KGCs working in the same groups G_1, G_2 and pairing $e : G_1 \times G_1 \to G_2$. Each KGC chooses its own master key s_i and publishes the public key $P_i = s_i P$. A user with identity ID registers with each of the n KGCs and receives n partial private keys $s_i Q_{ID}$ for the same ID. Then he computes his new private key by summing up these partial private keys:

$$\begin{aligned} D_{ID} &= s_1 Q_{ID} + s_2 Q_{ID} + \cdots + s_n Q_{ID} \\ &= (s_1 + s_2 + \cdots + s_n) Q_{ID}. \end{aligned}$$

In encryption $\sum_{i=1}^{n} P_i$ is used as the public key of the encryption scheme.

This approach is beautiful in theory and the private key of the user will be protected against collusion by up to $n-1$ out of the n KGCs. However, this approach has several problems.

First, the n key issuing operations are independent of each other and the key addition is computed by the user. Therefore any central organization of key issuing policy among the KGCs is difficult. In the real world, KGCs will issue a private key only when they are sure that the user will use the key correctly under their instructions (guidelines). If multiple KGCs have different instructions in the usage of keys, there can be argument for the usage of the new key computed by the user.

Second, in this approach the n KGCs have no control on the private keys and no countermeasure against user's illegal usage of keys, once they issued the private keys to the user. (Chen et al. 2002) even considered that the user can produce 2^n different public/private key pairs given only n KGCs.

Third, because of the simple arithmetical relation of the new private key and their parallel nature, an attacker can try to get a partial private key of a user by independently attacking each KGC, which can be added later to recover the real private key.

Fourth, the n KGCs have to check user's identification independently as the case of the threshold key issuing, which is quite a burden.

3.3 Certificate-based Encryption

(Gentry 2003) proposed a certificate-based encryption scheme which provides secure key issuing by using user-chosen secret information in a novel way. Assume that a KGC (or CA) has a master key s_0 and a public key $P_0 = s_0 P$. A user with identity ID requests the KGC to issue a private key. Then the scheme runs as follows.

- User chooses a secret $x_{ID} \in_R Z_q^*$ and computes his public key $Y_{ID} = x_{ID} P$. He sends Y_{ID} and ID to the KGC (Y_{ID} is published) and requests him to issue a certificate.

- KGC checks the identification of the user. He computes

  $$Info_{ID} = (Y_{ID}, ID),$$

  $$P_{ID} = H_1(P_0, i, Info_{ID}),$$

  where i denotes a time period of validity. Finally he computes a certificate

  $$Cert_{ID} = s_0 P_{ID}$$

  and gives it to the user ($Cert_{ID}$ is published).

- Receiving $Cert_{ID}$, user computes

  $$P'_{ID} = H_1(Info_{ID}),$$

  $$S_{ID} = Cert_{ID} + x_{ID} P'_{ID}.$$

  S_{ID} is a decryption key of the user.

- To encrypt a message m for the user, a sender computes

  $$P'_{ID} = H_1(Info_{ID}),$$

  $$P_{ID} = H_1(P_0, i, Info_{ID}),$$

  and

  $$g = e(P_0, P_{ID}) e(Y_{ID}, P'_{ID}).$$

  Choosing a random number $r \in_R Z_q^*$, he computes a ciphertext

  $$C = (U, V) = (rP, m \oplus H_2(g^r)).$$

- To decrypt the ciphertext $C = (U, V)$, the user recovers the plaintext as $V \oplus H_2(e(U, S_{ID})) = m$.

This scheme provides a secure key issuing successfully avoiding the key escrow problem. Moreover secure channel is not required; $Cert_{ID}$ can be sent over a public channel or published. Instead it becomes a certificate-based scheme losing the advantage of the ID-based schemes. It also inherits the revocation problem of the certificate-based schemes, therefore this scheme tries to use a short-lived private key with a time index i. From KGC a user obtains a publishable certificate $Cert_{ID}$ instead of a secret private key. Without the certificate a sender cannot be sure whether Y_{ID} is a correct public key of the receiver.

3.4 Certificateless Public Key Encryption

To solve the problem of (Gentry 2003), (Al-Riyami & Paterson 2003) proposed a new scheme which successfully removed the necessity of certificate, thus they called it certificateless public key cryptography. It is conceptually similar to the self-certified key (Girault 1991, Petersen & Horster 1997) because it uses user-chosen secret information in a similar way. Here we review their alternative key issuing method which provides binding property.

Assume that a KGC (or CA) has a master key s_0 and a public key $P_0 = s_0P$. A user with identity ID requests the KGC to issue a private key. Then the scheme runs as follows.

- User picks a random secret value $x_{ID} \in_R Z_q^*$ and computes a public key

  $$P_{ID} = (X_{ID}, Y_{ID}) = (x_{ID}P, x_{ID}P_0).$$

  He sends the public key P_{ID} and ID to the KGC and requests him to issue a private key.

- KGC checks the identification of the user. He extracts a partial private key by

  $$D'_{ID} = s_0Q_{ID},$$

  where $Q_{ID} = H_1(ID, P_{ID})$, and gives D'_{ID} to the user.

- User sets his private key as $D_{ID} = x_{ID}D'_{ID}$.

- To encrypt a message m for the user, a sender first computes $Q_{ID} = H_1(ID, P_{ID})$. Then choosing a random number $r \in_R Z_q^*$, he computes a ciphertext as

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

- To decrypt the ciphertext $C = (U, V)$, the user recovers the plaintext as

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

Although this scheme achieves key privacy, it provides only implicit authentication. A sender cannot be sure whether $P_{ID} = (X_{ID}, Y_{ID})$ is a correct public key of the receiver. He will be assured only after a successful communication.

4 Our Key Issuing Model

In this paper we propose a new secure key issuing protocol which preserves the advantages of ID-based cryptography. We assume a single key generation center (KGC) and multiple key privacy authorities (KPAs). Key privacy is provided by using multiple KPAs. To provide a secure channel between users and authorities a simple blinding technique is used in pairing-based cryptography.

4.1 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. Correct identification of user is as important as correct signing. The single authority approach is easy to implement cryptographically in ordinary PKI-based schemes, but it suffers from key escrow problem when implemented in ID-based cryptography. It seems to be inevitable to use multiple authorities to avoid key escrow problem.

In the real world there is an example of non-governmental organizations (NGOs) who are organized by themselves, not by the government, and trying to observe (or supervise) whether the government might do 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 roles of NGOs are very important. Although they do not have a legally approved authority like governmental organizations, they can provide some service of preventing 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 procedures, but major political parties dispatch observers to the voting office to prevent any illegal activity; illegal voters, double voting, threatening, miscounting, etc. Since each political party has different interests in the election, it is hard to assume that all observers collude. The supervision of voting and counting procedures 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 authorities like NGOs or observers (KPAs in this paper) who can provide services to keep the privacy of user's private key in ID-based cryptosystems, then there is a way that a single KGC and multiple KPAs can issue the ID-based private key in a secure manner. If this is possible, ID-based cryptography will become more useful in the real world.

4.2 Overview

In this proposal we introduce a single KGC and multiple KPAs. The key issuing process consists of the following three stages.

1. In key issuing stage, a user sends his identity and blinding factor to the KGC and requests him to issue a partial private key. Then, after checking the identity of the user, the KGC issues a partial private key to the user in a blinded manner.

2. In key securing stage, the user requests multiple KPAs in a sequential manner to provide key privacy service, then KPAs return the real private key in a blinded manner.

3. Finally, in key retrieving stage, the user unblinds it to retrieve the real private key.

Assuming the honesty of at least one KPA, the privacy of the private key is kept. Only the legitimate user who knows the blinding parameter can unblind the message to retrieve the private key. The proposed secure key issuing protocol overcomes the key escrow problem of ID-based cryptography, thus it can be applied to more complex applications satisfying stronger security requirements.

4.3 Entities and Their Roles

The entities participating in the secure key issuing protocol and their roles are as follows.

- KGC (or CA): A single KGC checks user's identification and issues a blinded partial private key to the user. Under a court order, he can be involved in the key escrow protocol.

- n KPAs: Multiple KPAs sequentially provide key privacy service to user's private key by issuing their signature in a blinded manner. We assume that at least one KPA will remain honest. Under a court order they can cooperate to provide a key escrow service for a specific message.
- User: He tries to get a private key of the ID-based cryptography through an interactive protocol with the KGC and n KPAs.

5 Proposed Secure Key Issuing Protocol

5.1 Key Issuing Protocol

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

Stage 1. System setup (by KGC)

As shown in Section 2, the 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 also picks his master key $s_0 \in Z_q^*$ at random and computes his public key $P_0 = s_0 P$. He publishes description of the groups G_1, G_2, the bilinear map e, hash functions H_1, H_2, H_3, and the public key P_0.

Stage 2. System public key setup (by KPAs)

The n KPAs establish their key pairs. For all $i = 1, \ldots, n$, KPA_i chooses his master key s_i and computes his public key $P_i = s_i P$. Then KPAs cooperate sequentially to compute the system public key

$$Y = s_0 s_1 \cdots s_n P,$$

which will be used as a system parameter in the group of users. More specifically,

$$KPA_1 \text{ computes } Y_1' = s_1 P_0,$$

$$KPA_2 \text{ computes } Y_2' = s_2 Y_1',$$

$$\cdots\cdots\cdots$$

$$KPA_n \text{ computes } Y_n' = s_n Y_{n-1}'.$$

Then $Y \equiv Y_n' = s_0 s_1 \cdots s_n P$ is published as the system public key. It will be used for encryption, signature verification, etc, by the users in the group. Note that the correctness of this sequential processes can be verified by $e(Y_i', P) \stackrel{?}{=} e(Y_{i-1}', P_i)$, where $Y_0' = P_0$.

Stage 3. Key issuing (by KGC and user)

A user with identity ID chooses a random secret x and computes a blinding factor $X = xP$. He requests the KGC to issue a partial private key by sending X and ID. Then the KGC issues a blinded partial private key as follows.

- Checks the identification of the user.
- Computes the public key of the user as
$$Q_{ID} = H_1(ID, KGC, KPA_1, \ldots, KPA_n).$$
- Computes a blinded partial private key as
$$Q_0' = H_3(e(s_0 X, P_0)) s_0 Q_{ID}.$$
- Computes KGC's signature on Q_0' as
$$Sig_0(Q_0') = s_0 Q_0'.$$
- Sends Q_0' and $Sig_0(Q_0')$ to the user.

Here $H_3(e(s_0 X, P_0))$ is a blinding factor; a secure channel between the user and the KGC. User can unblind it using his knowledge of x, since

$$H_3(e(s_0 X, P_0)) = H_3(e(s_0 x P, P_0)) = H_3(e(P_0, P_0)^x).$$

Stage 4. Key securing (by user and KPAs)

The user requests KPA_i $(i = 1, \ldots, n)$ sequentially to provide key privacy service by sending ID, X, Q_{i-1}', and $Sig_{i-1}(Q_{i-1}')$. Then KPA_i

- Checks $e(Sig_{i-1}(Q_{i-1}'), P) \stackrel{?}{=} e(Q_{i-1}', P_{i-1})$.
- Computes $Q_i' = H_3(e(s_i X, P_i)) s_i Q_{i-1}'$ and $Sig_i(Q_i') = s_i Q_i'$.
- Sends Q_i' and $Sig_i(Q_i')$ to the user.

He proceeds this process to KPA_n. Finally he receives

$$Q_n' = H_3(e(s_n X, P_n)) s_n Q_{n-1}'.$$

Stage 5. Key retrieving (by user)

The user retrieves his private key D_{ID} by unblinding Q_n' as follows.

$$\begin{aligned} D_{ID} &= \frac{Q_n'}{H_3(e(P_0, P_0)^x) \cdots H_3(e(P_n, P_n)^x)} \\ &= s_0 s_1 \cdots s_n Q_{ID} \end{aligned}$$

The user can verify the correctness of his private key by $e(D_{ID}, P) \stackrel{?}{=} e(Q_{ID}, Y)$.

The private key D_{ID} is a real ID-based private key corresponding to the public key Q_{ID} when $Y = s_0 \cdots s_n P$ is used as the system public key. Therefore this key pair can be used for any ID-based cryptosystems, such as encryptions (Boneh & Franklin 2001), signatures (Boneh, Lynn & Shacham 2002), etc.

5.2 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 encrypted 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 cooperatively decrypt the ciphertext to recover the plaintext m. Each entity can do the following computation sequentially,

$$((((e(Q_{ID}, U)^{s_1})^{s_2}) \cdots)^{s_n})^{s_0} = e(D_{ID}, U),$$

which will successfully recover the plaintext m. The correctness of this key escrow process can be verified by using the bilinear property. Note that this is a decryption per ciphertext, not a recovery of the user's private key.

5.3 Analysis

Since the private key of a user is computed cooperatively by the KGC and n KPAs, the privacy of user's private key is kept if at least one authority remains honest. Only the legitimate user who knows the blinding parameter can unblind the message to retrieve the private key.

The proposed secure key issuing protocol successfully overcomes the key escrow problem of ID-based cryptography, thus it can be applied to more complex applications satisfying stronger security requirements. Compared with (Gentry 2003, Al-Riyami & Paterson 2003), the proposed secure key issuing protocol issues a real ID-based private key, thus it can be used with any ID-based cryptosystems such as encryptions, signatures, and key agreements, preserving the advantages of ID-based cryptography.

Compared with the parallel composition model of (Chen et al. 2002), the proposed scheme uses a serial key issuing model. In (Chen et al. 2002), because of the simple arithmetical relation of the new private key and their parallel nature, an attacker can try to get partial private keys of a user by independently attacking each KGC, which can be added later to recover the real private key. But in our model partial collusion gives no useful information to the attacker. To obtain a useful information, all authorities have to participate in a sequential way.

Compared with the threshold key issuing, this scheme distributes the roles of user identification and key securing into KGC and KPAs, respectively, and they contribute to key generation in a sequential way.

In the point of efficiency, this scheme is less efficient than (Chen et al. 2002), since all the computation is done in a sequential way. But it reduces the cost of user identification. Furthermore, it provides solution to the problems discussed in Section 3.2.

6 Conclusion

In this paper we proposed a new secure key issuing protocol using the multiple authority approach. We proposed a new model of separating an authority into multiple parties; a single KGC who has the authority of user identification and key issuing and multiple KPAs who provide key privacy service. We can find similar analogies in the real world such as NGOs and electronic voting.

In cryptography the proposed method is a new approach to divide an authority into multiple parties. This approach will be useful in many applications where the authority is really powerful and some kind of control or observation is required. It will help to make the real world authorities be more distributed ones.

7 Acknowledgements

We acknowledge the support of the Australian government through ARC Linkage-International fellowship scheme 2003, Grant No: LX0346868.

References

Al-Riyami, S. & Paterson, K., 'Certificateless public key cryptography', Advances in Cryptology – Asiacrypt'2003, LNCS, Springer-Verlag, to appear.

Boneh, D., & Franklin, F., 'Identity-based encryption from the Weil pairing', Advances in Cryptology – Crypto'2001, LNCS 2139, Springer-Verlag, pp. 213–229.

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

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

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

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

Girault, M., 'Self-certified public keys', Advances in Cryptology - Eurocrypt'91, LNCS 547, Springer-Verlag, pp. 490–497.

Hess, F., 'Efficient Identity Based Signature Schemes Based on Pairings', Selected Areas in Cryptography – SAC 2002, LNCS 2595, Springer-Verlag, pp. 310–324.

Paterson, K., 'Cryptography from pairings: a snapshot of current research', Information Security Technical Report, Vol. 7(3), pp. 41–54.

Petersen, H. & Horster, P., 'Self certified keys - Concepts and Applications', Proc. Communications and Multimedia Security'97, Chapman & Hall, pp. 102–116.

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