An Enhanced Secure Key Issuing Protocol

Duc-Liem Vo* and Kwangjo Kim*

*International Research center for Information Security (IRIS),

Information and Communications University (ICU)

R504, 103-6, Munji-dong, Yuseong-gu, Daejeon 305-732, Republic of Korea

Abstract

Key escrow can be one of key limitations in ID-based cryptography since Key Generation Center is able to control all the critical information, particularly the private keys of all users. Several solutions in the open literature were suggested to solve this problem but they still have drawbacks such as requiring more computation or making ID-based cryptography more complicated. In this work, we propose a new secure key issuing (SKI) protocol including key escrow problem for ID-based cryptography using bilinear pairings. By defining the concrete security requirements for SKI protocol, we show that our protocol are more secure than the previous schemes.

I. Introduction

In the ID-based cryptography, a Key Generation Center (KGC) takes the full responsibility to generate the private key for each and all entities using their identity information. It is required that the KGC must be highly trustful due to its capability of decrypting any ciphertext or signing any message without the permission of an entity. As a result, the KGC needs to commit an agreement so called key escrow on using the entity's private to prevent any abuse of the key.

There are a variety of solutions to deal with this problem such as multiple authorities approach [3, 5] where the KGC's master private key is distributed to multiple authorities using threshold scheme; certificated-based [6] and certificateless PKC [1] approaches which let users choose an additional secret information by themselves. Recently, Lee *et al.* [9] proposed a secure key issuing protocol (SKI), through which, the trust level of KGC is spreading into multiple trusted third parties called Key Privacy Authorities (KPAs). This approach does not require user's multiple identifications; however, Lee *et al.*'s scheme is inefficient in terms of performance as well as not secure against various attacks such as impersonation, incompetency of KPAs [8]. Another scheme was proposed by Sui [10] but it requires multiple identifications and is vulnerable under insider attack and incompetency of KGCs [8]. Gangisishetti *et al.* [7] also proposed another version of SKI protocol working in parallel model.

In this work, we propose a new SKI protocol by using mediated security and analyze it comparatively with the previous schemes.

Organization. We brief some concepts of bilinear pairings and define requirements

for a SKI protocol in Section 2. Section 3 describes a new SKI protocol and its analysis in terms of security and performance is followed by Section 4. Supporting revocation for the protocol is discussed in Section 5 before making the concluding remarks in Section 6.

II. Background and Requirements

1. Bilinear Pairings

We summarize in mathematical notations some concepts of bilinear pairings which can be constructed from Weil or Tate pairings on supersingular elliptic curves.

Let G_1 and G_2 be additive and multiplicative groups of the same prime order q, respectively. Let P be a generator of G_1. Assume that the discrete logarithm problems in both G_1 and G_2 are hard. Let $e: G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following properties:

1. Bilinear: $e(aP,bP') = e(P,P')^{ab}$ for all P, $P' \in G_1$ and all $a,b \in Z$.

2. Non-degenerate: For all $P' \in G_1$ if $e(P,P')=1$ then $P=O$.

3. Computable: There is an efficient algorithm such as [3] to compute $e(P,P')$ for any P, $P' \in G_1$.

Group G_1 is called Gap Diffie-Hellman (GDH) group if in this group, the Computational Diffie-Hellman (CDH) problem is hard but the Decision Diffie-Hellman (DDH) problem is easy.

2. Requirements

Although many schemes were proposed, none of them have provided concrete requirements for a SKI protocol in ID-based cryptography. The SKI protocol should satisfy the following requirements:

Security A SKI protocol is required to be secure against impersonation attack so that an active adversary, who is capable of listening and modifying the protocol messages, could not actively modify the protocol messages without the users' detection.

Protection against insider attack With the assumption that the KGC is highly trustful and at least one KPA is honest, no one can gain advantage, *i.e.*, getting authorized on illegal document, when requesting services.

Robustness A SKI protocol can complete successfully in an event that some of KPAs are colluded.

There are the following entities participating in the SKI protocol for ID-based cryptography:

- KGC: The only highly trustful entity is responsible for checking user identity and issuing a blinded partial private key to the user. We assume that, the KGC is highly trustful entity and behaves honestly.

- n KPAs: Provide privacy service to user's private key and can operate in t-out-of-n secret sharing scheme to provide robustness when up to $n-t$ KPAs are not available.

- User: Interact with KGC and KPAs in order to get a private key for ID-based cryptosystems securely.

III. Proposed Scheme

1. Overview

As described above, our scheme has a KGC generating private keys and n KPAs providing privacy services for users. However, unlike the previous schemes, we utilize the concept of mediated security [2] in the construction of our scheme. At a glance on this concept, a KGC creates a

private key for a user and divides the key into two parts: the first part goes to the user and the second part goes to a so-called security mediator (SEM), an online entity providing security services, *i.e.*, decryption and digital signature generation, for users. Whenever the user needs security services, he has to communicate with a SEM, otherwise he can do nothing with his private key. The advantage of mediated security is the immediate revocation of public keys to be feasible.

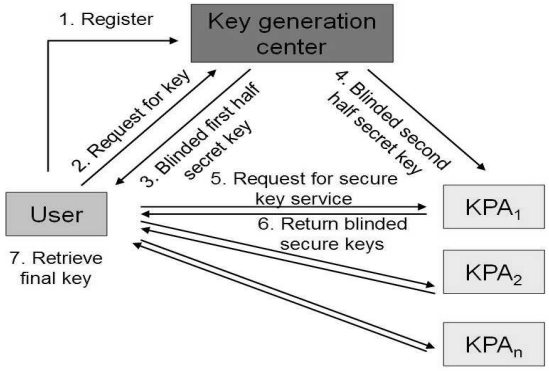

Fig 1: Secure Key Issuing Protocol overview

In our proposed scheme, we let the KGC divide a user's private key into two parts and let KPAs act similarly to a SEM. After generating and dividing a private key for a user, the KGC gives the first part to the user blindly while keeping secretly the second part. When the user requests securing key service for his first part of the private key to KPAs, the KPAs examine the user's request and ask the KGC for the corresponding part of the user's private key. After gathering and checking the user's second part of the private key, KPAs will provide securing key service on both parts and deliver to the user if two parts match, otherwise, KPAs will ignore the user's request. A valid user ultimately will get all necessary information to form his private key after communicating with all KPAs. He processes these steps to get his private key and then uses it without limitation. This differs from a mediated security approach where a user needs to communicate with a SEM whenever he wants to create a digital signature or decrypt a ciphertext.

2. Detailed Protocol

The SKI protocol consists of the following phases: system setup, system public key setup, key issuing protocol, key securing and key retrieving phases.

■ **System setup phase**. The KGC runs IG with the security parameter k as input to generate group G_1 and G_2 of prime order q, a generator P of G_1, a bilinear map $e:G_1\times G_1\rightarrow G_2$, and hash functions $H_1:\{0,1\}^*\rightarrow G_1$.

■ **System public key setup phase**. The KGC picks a master key $s_0\in_R Z_q^*$ and computes its public key $P_{kgc}=s_0P$. n KPAs pick their private keys and compute the corresponding public key $s_i\in_R Z_q^*$ $P_i=s_iP$, for $i=1,2,...,n$. Each KPA$_i$ computes $Y'_i=s_iP_{kgc}$ and sends it to the KGC. The KGC computes the system public key $Y=\sum_{i=1}^{n} Y_i=s_0(s_1+s_2+...+s_n)P$. The KGC publishes system parameters as $\langle G_1, G_2, q, e, H_1, n, P, P_{kgc}, P_1,P_2,..., P_n,Y\rangle$.

A user, identified by ID, registers in advance his credentials $\langle ID, R=rP\rangle$ with KGC, where $r\in_R Z_q^*$ was chosen secretly by the user. The credentials are also available for KPAs.

■ **Key issuing phase**. A registered user, identity ID, interacts with the KGC to request the private key. The user computes $Q_{ID}=H_1(ID)$, $D_{ID}=rQ_{ID}$, and sends the tuple $\langle ID, D_{ID}\rangle$ to the KGC. The KGC performs:

1. Search for the tuple $\langle ID, R\rangle$; check the validity of the tuple $\langle ID, D_{ID}\rangle$ by verifying the equation $e(D_{ID},P)= e(Q_{ID},R)$.

2. Pick randomly $s_{0,u}\in Z_q^*$ and compute $s_{0,KPA}= s_0-s_{0,u} \bmod q$.

3. Compute $Q'_0=s_{0,u}D_{ID}$, $T=e(s_{0,KPA}Q_{ID},R)$, and send these to the user. The tuple $\langle ID, W'_0=s_{0,KPA}D_{ID}\rangle$ is sent to the KPAs.

Receiving data from the KGC, the user verifies if this data is correct or not by checking $e(Q'_0,P)T=e(D_{ID},P_{kgc})$. This equation must hold since $rP_{kgc}=s_0R=s_0rP$.

■ **Key securing phase.** The user can contact KPAs simultaneously in order to request the key securing service.

1. The user sends the tuple $\langle Q'_0, D_{ID}\rangle$ along with his identity ID to KPAs.

2. Each KPA gets a tuple $\langle ID, W'_0=s_{0,KPA}D_{ID}\rangle$ and checks if $e(D_{ID},P_{kgc})=e(Q'_0+W'_0,P)$ to verify the right user requests services.

3. KPA$_i$ provides key securing service by computing: $Q_i = s_iQ'_0$, $W_i = s_iW'_0$, $P'_i=s_iP_{kgc}$. KPA$_i$ sends back $\langle Q_i,W_i,P'_i\rangle$ to the user.

Receiving data from KPA$_i$, the user checks if the following equalities hold: $e(Q_i,P)=e(Q'_0,P_i)$,and $e(Q_i+W_i,P)=e(D_{ID},P'_i)$.

■ **Key retrieving phase.** When the user receives correct data from all KPAs, he can compute the private key as follows:

$$S_{ID} = r^{-1}\sum_{i=1}^{n}(Q_i+W_i)=(s_1 + s_2 +...+ s_n)s_0Q_{ID}$$

The user can verify the secret key by checking the equation: $e(S_{ID}, P)=e(Q_{ID}, Y)$

IV. Security Analysis

As mentioned before, our protocol requires the KGC to be highly trustful, that means the KGC never colludes with the KPAs to perform illegal activities. Therefore, our proposed protocol keeps the user's privacy if at least one of the KPAs remains honest. Our protocol satisfies the following properties:

Security. In the proposed protocol, the user communicates with the KGC in an authenticated way. The user proves to the KGC his possession of the secret r, which also binds to his identity when communicating with the KGC. In the key securing phase, although KPAs know the other part of the user's private key from the KGC, no KPA has any advantage in discovering the whole parts of the user's private issued by the KGC since the user only sends the blinded key to KPAs. KPAs can just check if the partial key sent by a user is associated with the user ID and the key's validity.

On the other hand, KPAs can ensure that only the user has information that matches with the one in the KGC's database have key securing service. The malicious users cannot impersonate the valid users requesting key securing services from KPAs without detection. This is one of advantages in our scheme compared with the previous works.

In addition, all protocol messages sent among parties are checked for integrity before accepted for further processing.

Protection against insider attack. Since we assume that the KGC is fully trusted, we consider insider attacks in our protocol as the malicious KPAs. The malicious KPAs can have access to the KGC to get a partial private key of the users. However, they cannot request secure service from the honest KPAs because the honest KPAs always check if there is a

correct partial private key provided from the "user" (the malicious KPA). The correct partial private key is kept by the legitimate users, and only blinding version are sent to KPAs. Therefore malicious KPAs could not gain any advantage in this case.

To provide robustness, our protocol can apply threshold cryptography. When using a t-out-of-n threshold scheme, n KPAs cooperate to create a shared private key, and it is required that there are at least t KPAs available in order to issue key securing services. Under this setting, the key issuing protocol allows up to $n-t$ KPAs can be corrupt or unavailable.

Algorithm		Ours	[7]	[9]
Setup		(2n+2)P+ 2nS+nA	(2n+2)P+ 2nS+nA	2nP+2nS
Issuing	KGC	3P+3S	2P+S	3P+6S+3h
	User	2P+2S	2P+2S	2P+S
Securing	KGC	2nP+3nS+nA	2nP+nS	3nP+4nS+nh
	User	4nP+nA	2nP	2nP
Retrieving		2P+S+(2n-1)A+I	2P+S+nA+I	(n+2)P+nS+nh+nI
Model		Parallel	Parallel	Sequential

Table 1: Model and computational complexity.

Table 1 shows the computational complexity of our protocol compared with others. P stands for the computational complexity of a pairing computation, S for a scalar multiplication and A for a point addition in an elliptic curve. h is hash operation and I is modular inversion computation.

As can be seen, our scheme requires more computation than [7]. Nevertheless, compared with [9] operating in sequential model, our scheme has shown advantages in performance. To improving performance, we can consider that if all KPAs work correctly, the verification of Q_i's can be omitted saving $2n$ pairing computation. The validity of those values can be checked later in the key retrieving phase by aggregated verification [4]. In the case this verification does not hold then we need to check back validity of Q_i's. This technique can also be applied to the system public key setup phase to save pairing computation. Moreover, our scheme can be extended easily to support revocation while [7] and [9] cannot. We describe details in the next section.

V. Revocation

Besides key escrow problem, revocation of the public keys of users in ID-based cryptography remains an open problem. Boneh [3] suggested a method adding a valid time period to the public key (identity). However, it is difficult to determine how long the period lasts for an efficient implementation. Employing a SEM [2], an ID-based cryptosystem can not only provide revocation easily but also eliminate users from using expired private keys for illegitimate purposes.

The simplest approach to enable our SKI protocol to support revocation is to include an online trusted third party acting as a SEM. The operations of SEM are described as follows:

- After creating and dividing a user's private key in two parts, KGC sends the first part to the user and the second to SEM. SEM keeps this part securely and gives to KPAs based on their requests.

- When a user requests for securing key services from KPAs, KPAs perform the steps as in Key securing phase but instead of sending back to the user the secured second parts of user's private key (i.e. W_i, for i=1, 2,...,n), KPAs send those parts to SEM. SEM verifies the data sent by KPAs and computes the complete second part of user's private key. This operation also

reduces work load for users. The first parts of user's private key is applied secure services by KPAs and sent to the user as normal.

- SEM, with the complete second part of user's private key, provide security services (decrypting ciphertext or issuing a signature) to the users. Before granting services, SEM checks the identity of the user and his revocation status. If the user's status is valid, he gets services from SEM, otherwise he gets nothing.

As can be seen, SEM makes immediate revocation of users' public key easily. It will be more efficient if we adapt threshold cryptography to our scheme in which a dealer of the threshold scheme for KPAs' sharing a secret key serves as the role of SEM. In this case, we can achieve robustness securing key services while providing revocation capability. However, it is inconvenient for users to contact with SEM all the time in order to get services.

VI. Concluding Remarks

Although ID-based cryptography simplifies key management and key distribution compared with the traditional PKC, ID-based cryptosystems are far from being practical due to key escrow problem. In this work, we have proposed an enhanced SKI protocol using multiple authorities working in parallel model. Our proposed protocol satisfies security requirements and can enable ID-based cryptography to be more practical. Moreover, by utilizing mediated security concept, we can extend the protocol for revocation capability. For further work, we would like to make our protocol more formal and improve its performance.

References

[1] S. Al-Riyami, and K. Paterson, "Certificateless Public Key Cryptography", Advances in Cryptology . Asiacrypt 2003, LNCS, Springer-Verlag,

[2] D. Boneh, X. Ding, G. Tsudik, and C. Wong, "A Method for Fast Revocation of Public Key Certificates and Security Capability," Proc. of the 10th USENIX Security Symposium, pp. 297-308.

[3] D. Boneh and M. Franklin, "ID-based Encryption from the Weil-pairing," Advances in Cryptology - Crypto'01, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[4] 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.

[5] L. Chen, K. Harrison, N. P. Smart, and D. Soldera, "Applications of Multiple Trust Authorities in Pairing Based Cryptosystems," InfraSec 2002, LNCS 2437, Springer-Verlag, pp. 260-275.

[6] C. Gentry, "Certificate-based Encryption and the Certificate Revocation Problem," Advances in Cryptology - Eurocrypt'03, LNCS 2656, Springer-Verlag, pp. 272-293.

[7] R. Gangishetti, M. C. Gorantla, M. L. Das, A. Saxena and V. P. Gulati, "An Efficient Secure Key Issuing Protocol in ID-Based Cryptosystems," In Proc. of the International Conference on Information Technology: Coding and Computing (ITCC 2005), Volume 1, IEEE Computer Society, pp. 674-678.

[8] R. Gangishetti, M. C. Gorantla, M. L. Das, and A. Saxena, "Cryptanalysis of Key Issuing Protocols in ID-based Crypto-systems," http://arxiv.org/abs/cs/0506015.

[9] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, S. Yoo, "Secure Key Issuing in ID-based Cryptography," In Proc. of the 2nd Australian Information Security Workshop. AISW'04, pp. 69-74.

[10] A. Sui, S. S. M. Chow, L. C. K. Hui, S. M. Yiu, K. P. Chow, W. W. Tsang, C. F. Chong, K.H. Pun, H. W. Chan, "Seperable and Anonymous Identity-Based Key Issuing without Secure Channel," Cryptology ePrint Archive. http://eprint.iacr.org/2004/322.