Certificateless Authenticated Key Distribution and Conference Key Distribution Schemes

Chien-Lung Hsu
Department of Information Management
Chang-Gung University,
#259 Wen-Hwa 1st Rd., Kwei-Shan Tao-Yuan 333, Taiwan
E-Mail: clhsu@mail.cgu.edu.tw

Tzong-Chen Wu
Department of Information Management
National Taiwan University of Science and Technology
#43, Sec.4, Keelung Rd., Taipei 106, Taiwan

Kuo-Yu Tsai
Taiwan Information Security Center
at National Taiwan University of Science and Technology
#43, Sec.4, Keelung Rd., Taipei 106, Taiwan

ABSTRACT
In this paper, we propose two authenticated key distribution protocols and one conference key protocol using identity-based, self-certified public keys. These proposed protocols have the following properties:

(i) No extra public key certificates and computational efforts are required for verification of authenticity for public keys.
(ii) Key establishment and validation of the desired public keys are integrated in a logically single step.
(iii) Each registering user can use his derived private key to check the validity of his public key issued by the system authority.
(iv) The system authority cannot access to users’ private keys. Hence, all private keys cannot be compromised in case of disruption or corruption of the system authority.
(v) The proposed protocols provide the security property of key authentication whereby one user is assured that no other user aside from specifically identified user(s) may gain access to the derived shared key/conference key.

Keywords: Certificateless, key distribution, conference key distribution
1. INTRODUCTION

In 1976, Diffie and Hellman invented a concept of the public key cryptosystem [3], in which each user has a private key and a public key. The private key is kept secret whereas the public key is made public in such a way that knowledge of the public key does not allow computation of the private key. Using such key pair, any two users could establish a common secret shared key without sharing the same key beforehand as a secret key cryptosystem. However, most of the public key cryptosystems suffer from so-called active attacks. For example, an active adversary may masquerade as some user to communicate or intercept the communication between two users by substituting a fake public key for the genuine one [5]. To avoid such attacks, it is necessary to authenticate public keys to achieve the data origin authentication so that any user can be convinced that he obtains the valid public key with respect to the legitimate user.

There are three different approaches for achieving public key authentication. They are: certificate-based [6, 9], identity-based [2, 11, 15, 16, 19, 20], and self-certified [5, 12, 14, 17, 18]. In addition, there is the certificateless approach [1, 10].

Certificate-based approach. In this approach, which is extensively adopted, every user has a key pair and an extra public key certificate. The public key certificate is to make the public key available to others such that its authenticity and validity are verifiable. In practice, X. 509 certificates are commonly used [8]. For this purpose, the system must require a trusted certification authority (CA) to issue and maintain such public key certificates associated with public keys. A public key certificate can be regarded as a signature of the user’s identity information, the user’s public key, the expiration time of the public key, and some public information under the private key of CA. To make public keys and certificates available to users, CA also should create and maintain a public key directory and a certificate revocation list (CRL). From the public key directory, users can obtain the desired public keys and the corresponding certificates. Looking up the CRL, users could ascertain whether the obtained certificates are valid. Extra computational effort is required in order for users to validate the authenticity of the desired public keys with valid certificates. Consequently, extra communication costs and computational efforts should be required to validate a public key.

Identity-based approach. This approach is inherited from the concept of the identity-based public key cryptosystem introduced by Shamir [15]. Every user does not have an explicit public key as before. The public key is replaced by his publicly available identity information, which can uniquely identify him or her and can be undeniably associated with that individual. The corresponding private key is computed from a one-way trapdoor function of some privileged information known only to the system authority (SA). The identity-based approach differs from the certificate-based approach in that the authenticity of users’ public keys is verified implicitly. Hence, it does not require extra effort...
and information for users to validate the authenticity of public keys. The main disadvantage of this system, however, is an unconditional trust in the SA, which therein has access to users’ private keys. Consider the scenario that, if the SA is disrupted or corrupted, all private keys might be compromised or revealed to illegitimate users.

**Self-certified approach.** In 1991, Girault [5] proposed another variation of the public key system — namely, the self-certified public key system — with implicitly self-certified public keys to resolve the verification of the authenticity of the public key. Under this approach, the private key of each user could be determined by the user or together by the user and the SA such that only the user knows the private key (as opposed to the identify-based approach in which the SA has access to each user’s private key). The so-called self-certified public key of each user is issued by the SA and its authenticity could be verified by the associated user only with his or her chosen corresponding private key. The verification of the authenticity of self-certified public keys can be further carried out with subsequent cryptographic application (e.g., a key distribution or signature scheme) in a logically single step without any extra certificate as in the certificate-based approach. This approach earns more efficiency in saving the communicational costs and the computational efforts to validate public keys, compared with the certificate-based approach.

**Certificateless approach.** Using ideas similar to the self-certified public key systems, Al-Riyami and Paterson [1] proposed a new approach; namely, certificateless public key cryptography (CL-PKC). In this approach, the SA generates a partial private key, and then each user generates his or her private key and public key using his or her secret value and partial private key. The concept was to oppose the SA’s having access to each user’s private key, as is the case in the identity-based approach, and was the absence of digital certificates and their important management overhead. However, CL-PKC cannot resolve the verification of the authenticity for the public key.

Elaborating on the merits inherent in both the identity-based and the self-certified approaches, Saeednia [12] proposed an identity-based, self-certified public key system and applied it to realize key exchange protocols. Later, Wu et al. [18] pointed out that Saeednia’s public key system was vulnerable to the impersonate attack and then presented an improvement to eliminate the flaw. Recently, based on the discrete logarithms, Wu [17] proposed a new identity-based, self-certified public key system and applied it to realize digital signature/multisignature schemes.

In this paper, we intend to propose two authenticated key distribution protocols by using identity-based, self-certified public keys and to apply them to construct a conference key distribution protocol. The shared key established by two communicating users in the first key distribution protocol is fixed and the same for every session initiated by the same users, but it is variable in the shared
key distribution protocol. In the conference key distribution protocol, the chairman could perform one of two key distribution protocols to establish the shared key with each participant, respectively. And then, he can broadcast the conference information such that only the legitimate participants can derive the conference key with the genuine shared keys. In all of the proposed protocols, the key establishment and verification of public key authentication are achieved simultaneously. Hence, they gain much efficiency in saving the communication costs and computational efforts incurred when these two tasks are performed independently and subsequently.

2. **SELF-CERTIFIED KEY DISTRIBUTION PROTOCOLS**

   This section discusses the system model of the self-certified key distribution protocols and the realization of the protocols.

2.1. **System Model of the Self-Certified Key Distribution Protocols**

   The self-certified key distribution protocol can be divided into three stages: system setup, user registration, and key distribution. The system requires a system authority (SA) responsible for defining system parameters and dealing with user registration.

   In the system setup stage, the SA generates the key pair that will be used in the latter time and defines system parameters.

   In the user registration stage, each user prepares his or her identity information such as the identifier, name, and address, and then performs a zero-knowledge proof protocol in collaboration with the SA to determine his or her self-certified private-key/public-key pair by randomly choosing a master key. Here, we elaborate an exponentiation function to construct the zero-knowledge proof protocol where computing the discrete logarithm over finite field is computationally infeasible [10]. Hence, the SA cannot obtain the master key from the registering commitment sent from the registering user. After completing the zero-knowledge proof protocol, the registering user will obtain his or her self-certified public key and witness for the public key sent from the SA via a secret channel. The user can derive his or her private key with the knowledge of the witness issued by the SA as well as the chosen master key. Meanwhile, the user can use the derived private key to check the validity of the self-certified public key issued by the SA. Thereafter, the self-certified key pair will be used in the key distribution stage. Note that the SA cannot derive the private key of the registering user and masquerade as the user because of the unknown master key. The user registration stage is depicted in Figure 1.
In the first key distribution protocol (abbreviated hereafter as KDP1), the shared key established by two communicating users is fixed and the same for every session initiated by the same users (Figure 2-a). In contrast to KDP1, the shared key established by two communicating users for every session is variable in the second key distribution protocol (abbreviated hereafter as KDP2). In KDP2, each communicating user randomly chooses an integer (protected by a discrete logarithm problem) and sends the corresponding commitment together with his or her identity information and self-certified public key to the other. They can use respective private keys to derive the shared key from the received messages. KDP2 is illustrated in Figure 2-b. Note that:

(i) Verification of respective public keys and the shared keys established in KDP1/KDP2 are implicitly integrated.
(ii) The identities of participants can be corroborated.
(iii) KDP2 is secure against the known-key attack (i.e., if the past shared key is compromised, then the adversary can compromise the future shared key or plot impersonation [6]), but KDP1 is not).

**Figure 1. The User Registration Stage**

**Figure 2-a & 2-b. Key Distribution Protocols**
2.2. Realization of Self-Certified Key Distribution Protocols

The system setup, user registration, and key distribution stages of the proposed self-certified key distribution protocols are stated below.

System setup. For system setup, the SA randomly selects two large primes $p$ and $q$, where $q | p - 1$, and a generator $\alpha$ of order $q$ over GF$(p)$. Meanwhile, the SA also chooses a one-way hash function $h$ that accepts a variable length input and produces a fixed length output. After that, the SA publishes $p$, $q$, $\alpha$, and $h$. The SA also determines a private-key/public-key pair $(\gamma, \beta)$, where $\gamma \in Z_q^*$ and $\beta = \alpha^\gamma \mod p$.

User registration. Let $u_i$ be the registering user and $I_i$ be the identity information with respect to $u_i$ such as the identifier, name, and address. $u_i$ sends a request to the SA and then collaborate with the SA to determine the self-certified private-key/public-key pair. First, $u_i$ computes a registering commitment $v_i$ as

$$v_i = \alpha^{-h(k_{II})} \mod p$$  \hspace{1cm} (1)

where $k_i \in Z_q$ is regarded as a secret master key. Then, $u_i$ transmits $\{I_i, v_i\}$ to the SA. Upon receiving $\{I_i, v_i\}$ transmitted from $u_i$, the SA checks whether $I_i$ has been registered. If $I_i$ has been registered, the SA must request $u_i$ to re-send new identity information and registering commitment until the chosen identity information has not been registered. Furthermore, the SA computes a self-certified public key $y_i$ and a witness $w_i$ with respect to $u_i$ as

$$y_i = v_i \cdot \alpha^{z_i} - h(I_i) \mod p$$  \hspace{1cm} (2)

$$w_i = z_i + \gamma \cdot (y_i + h(I_i)) \mod q$$  \hspace{1cm} (3)

where $z_i \in Z_q$. Then, the SA sends $\{y_i, w_i\}$ back to $u_i$. With the knowledge of $w_i$, $u_i$ can derive the private key $x_i$ from the following equation:

$$x_i = w_i - h(k_{II} \parallel I_i) \mod q$$  \hspace{1cm} (4)

To verify the authenticity of the key pair $\{x_i, y_i\}$, $u_i$ checks the following equality:

$$\alpha^{x_i} \mod p = (y_i + h(I_i)) \cdot \beta^{y_i + h(I_i)} \mod p$$  \hspace{1cm} (5)

With the above equality holding, $(x_i, y_i)$ will be regarded as a valid self-certified private-key/public-key pair of $u_i$. Note that the private key $x_i$ is unknown to the SA because of the unknown master key $k_i$. 

Volume 5, Number 1, June 2010
Theorem 1. The authenticity of the key pair \((x_i, y_i)\) with respect to \(u_i\) will be verified correctly by Eqn. (5).

Proof: Substituting Eqn. (3) into Eqn. (4), we have:

\[ x_i = z_i + \gamma \cdot (y_i + h(I_i)) - h(k_i || I_i) \mod q \]

Raising both sides of the above equation to exponents to the base \(\alpha\), we have

\[ \alpha^{x_i} \mod p = \alpha^{z_i + \gamma \cdot (y_i + h(I_i)) - h(k_i || I_i)} \mod p \]
\[ = \alpha^{z_i} \cdot v_i \mod p \]
\[ = \alpha^{z_i} \cdot \beta^{y_i + h(I_i)} \cdot v_i \mod p \]
\[ = (y_i + h(I_i)) \cdot \beta^{y_i + h(I_i)} \mod p \]

which implies Eqn. (5).
Q.E.D.

Key distribution protocols. Let \(u_i\) and \(u_j\) be two users who want to establish a secret key shared between them by applying either KDP1 or KDP2. Details of these two protocols are described as follows.

KDP1: Each user transmits his or her identity information and public key to the correspondent – i.e., \(u_i\) transmits \(I_i, y_i\) to \(u_j\) and \(u_j\) sends \(I_j, y_j\) to \(u_i\) as shown earlier in Figure 2. Thereafter, \(u_i\) and \(u_j\) can compute the shared key from the following equality, respectively:

\[ SK_{ij} = ((y_j + h(I_j)) \cdot \beta^{y_j + h(I_j)})^{x_i} \mod p \]  \hspace{1cm} (6)
\[ SK_{ji} = ((y_i + h(I_i)) \cdot \beta^{y_i + h(I_i)})^{x_j} \mod p \]  \hspace{1cm} (7)

\(u_i\) and \(u_j\) can check whether \(SK_{ij} = SK_{ji}\) in a simple way as [5]. Note that the equivalence of \(SK_{ij}\) and \(SK_{ji}\) also implies that the authenticity of \(y_i\) and \(y_j\) are verified, and the identities of \(u_i\) and \(u_j\) are corroborated. The correctness of KDP1 is showed as follows. From Eqn. (6), we have:

\[ SK_{ij} = ((y_j + h(I_j)) \cdot \beta^{y_j + h(I_j)})^{x_i} \mod p \]
\[ = ((v_j \cdot \alpha^{z_j}) \cdot \beta^{y_j + h(I_j)})^{x_i} \mod p \] \hspace{1cm} \text{by Eqn. (2)}
\[ = ((v_j \cdot \alpha^{z_j}) \cdot \alpha^{\gamma(y_j + h(I_j))})^{x_i} \mod p \]
\begin{align*}
  & (\alpha^{-h(k_jI_j)} \cdot \alpha^{z_j}) \cdot \alpha^{p(y_j + h(I_j))} x_i \mod p \quad \text{by Eqn. (1)} \\
  & = (\alpha^{w_j-h(k_jI_j)} x_i) \mod p \quad \text{by Eqn. (3)} \\
  & = \alpha^{x_j \cdot x_i} \mod p \quad \text{by Eqn. (4)}
\end{align*}

Eqn. (7) can also be rewritten as \( SK_{ji} = \alpha^{x_j \cdot x_j} \mod p \), which implies any two communicating users can share a secret key by knowing the identity information and his or her correspondent’s public key.

**KDP2:** To establish a shared key, \( u_i \) computes \( t_i = \alpha^{r_i} \mod p \), where \( r_i \in \mathbb{Z}_p \), and sends \( \{I_i, y_i, t_i\} \) to \( u_j \). Meanwhile, \( u_j \) also computes \( t_j = \alpha^{r_j} \mod p \), where \( r_j \in \mathbb{Z}_q \), and sends \( \{I_j, y_j, t_j\} \) to \( u_i \). By Eqns. (10) and (11), \( u_i \) and \( u_j \) compute the shared key, respectively:

\begin{align*}
  SK_{ij} &= ((y_j + h(I_j)) \cdot \beta^{y_j + h(I_j)}) \cdot t_j \cdot x_i \mod p \quad \text{(10)} \\
  SK_{ji} &= ((y_i + h(I_i)) \cdot \beta^{y_i + h(I_i)}) \cdot t_j \cdot x_i \mod p \quad \text{(11)}
\end{align*}

Note that, \( t_i \) and \( t_j \) are variable according to time-variant parameters \( r_i \) and \( r_j \). The shared key \( SK_{ij} \) (or \( SK_{ji} \)) is variable and different for each session since they are computed from \( t_i \) and \( t_j \). From Eqns. (10) and (11), if \( SK_{ij} \) is equal to \( SK_{ji} \), then it implies that the authenticity of public keys \( y_i \) and \( y_j \) are verified. The correctness of the protocol is shown as follows.

\begin{align*}
  SK_{ij} &= ((y_j + h(I_j)) \cdot \beta^{y_j + h(I_j)}) \cdot t_j \cdot x_i \mod p \\
  &= ((v_i \cdot \alpha^{z_j}) \cdot \beta^{y_j + h(I_j)}) \cdot t_j \cdot x_i \mod p \quad \text{by Eqn. (2)} \\
  &= ((v_i \cdot \alpha^{z_j}) \cdot \alpha^{p(y_j + h(I_j))}) \cdot t_j \cdot x_i \mod p \\
  &= ((\alpha^{-h(k_jI_j)} + z_j) \cdot \alpha^{p(y_j + h(I_j))}) \cdot t_j \cdot x_i \mod p \quad \text{by Eqn. (1)} \\
  &= (\alpha^{w_j-h(k_jI_j)}) \cdot t_j \cdot x_i \mod p \quad \text{by Eqn. (3)} \\
  &= \alpha^{x_j \cdot r_i} \cdot t_j \cdot x_i \mod p \quad \text{by Eqn. (4)} \\
  &= \alpha^{x_j \cdot r_i} \cdot \alpha^{r_j \cdot x_i} \mod p \quad \text{by Eqn. (9)}
\end{align*}

Eqn. (11) can also be written as \( SK_{ji} = \alpha^{x_j \cdot r_j} \cdot \alpha^{r_j \cdot x_j} \mod p \) which implies Eqn. (10).
3. SELF-CERTIFIED CONFERENCE KEY DISTRIBUTION PROTOCOL

We apply KDP1/KDP2 stated in the previous section to construct a conference key distribution protocol, called a self-certified conference key distribution protocol.

3.1. The System Model of the Self-Certified Conference Key Distribution Protocol

The conference key distribution protocol is divided into four stages: system setup, user registration, conference key distribution, and conference key recovery. The system setup and user registration stages are the same as those described in the previous section. When $n$ participants want to hold a secret conference, any of them can act as a chairman to perform conference key distribution. In the conference key distribution stage, the chairman randomly chooses a conference key and then applies KDP1 or KDP2 to establish a shared key between the chairman and other participants. Further, he or she constructs an authentic message polynomial with the knowledge of the chosen conference key, the established shared keys, and participants’ identity information by Lagrange polynomial formula [6] so that only legal participants can recover the conference key. Finally, the chairman broadcasts his or her identity information, self-certified public key, and the constructed authentic message polynomial to all participants of the conference. Thereafter, legal participants can perform conference key recovery stage to recover the conference key. The self-certified conference key distribution protocol is shown in Figure 3.

![Figure 3. A Self-Certified Conference Distribution Protocol](image-url)
3.2. Realization of the Self-Certified Conference Key Distribution Protocol

We will apply KDP1 or KDP2 as described in Section 2 to establish shared keys between the chairman and each participant. By using the established shared keys, the conference key protocol can be constructed by all participants. For simplicity, we only describe the details of the conference key distribution and conference recovery stages.

**Conference key distribution.** Let \( G = \{u_0, u_1, \ldots, u_n\} \) be the set of users in this system, \( P \) the set of the participants in \( G \), and \( \overline{P} \) the set of the non-participants in \( G \) where \( \overline{P} = G \setminus P \). Without loss of generality, we assume \( u_0 \) be the chairman of the conference. First, \( u_0 \) initiates a secret conference and performs the steps to broadcast the established authentic message to \( P \):

1. Apply KDP1 or KDP2 to compute \( SK_{0i} \) as the shared key between \( u_0 \) and \( u_i \in P \).

2. Randomly choose a conference key \( CK \in Z_q \).

3. Compute \( |P| \) points as:

\[
(h(I_i), CK \oplus h(SK_{0i} || I_0 || I_i))
\]

for \( i = 1, 2, \ldots, |P| \), where \( |P| \) stands for the number of users in the set \( P \). And then, \( u_0 \) constructs a (\( |P| - 1 \))-degree polynomial \( Q_1(x) \) with Lagrange formula as:

\[
Q_1(x) = \sum_{i=1}^{|P|} ((CK \oplus h(SK_{0i} || I_0 || I_i)) \cdot L_i) \mod p
\]

where \( L_i = \prod_{j=1, j\neq i} (x - h(I_j))(h(I_i) - h(I_j))^{-1} \mod p \).

4. Construct a polynomial with degree \( |P| \) as:

\[
Q_2(x) = \prod_{u_i \in P} (x - h(I_i)) \mod p
\]

5. Randomly choose an integer \( c \in Z_p \) such that

\[
Q_1(h(I_j)) + c \cdot Q_2(h(I_j)) \neq CK \oplus h(SK_{0j} || I_0 || I_j) \mod p
\]

for any \( u_j \in \overline{P} \).

6. Compute the authentic message polynomial:

\[
Q(x) = Q_1(x) + c \cdot Q_2(x) \mod p
\]

and broadcast \( \{I_0, y_0, Q(x)\} \) to \( G \).
Conference key recovery. Each participant \( u_i \in P \) derives the shared key \( SK_{0i} \) shared between him or her and the chairman with KDP1 or KDP2. Then, the participant computes \( Q(h(I_j)) \) using his or her identity information \( I_j \) and the authentic message polynomial \( Q(x) \), where

\[
Q(h(I_j)) \mod p = Q_1(h(I_j)) + c \cdot Q_2(h(I_j)) \mod p = CK \oplus h(SK_{0i} \parallel I_0 \parallel I_j),
\]

\[
Q_1(x) = \sum_{i=1}^{|P|} ((CK \oplus h(SK_{0i} \parallel I_0 \parallel I_j)) \cdot L_i) \mod p,
\]

\[
Q_2(x) = \prod_{u_i \in P} (x - h(I_j)) \mod p.
\]

Finally, \( u_i \in P \) can compute the conference key \( CK \) as \( CK = Q(h(I_j)) \oplus h(SK_{0i} \parallel I_0 \parallel I_j) \) by using \( SK_{0i} \) and \( Q(h(I_j)) \). Each \( u_i \in P \) can easily prove that the proposed protocol works correctly. Since only the genuine \( SK_{0i} \), established by KDP1 or KDP2, can be used to derive \( CK \), participants’ public keys and identities which are corroborated as discussed in Section 2. Note that any \( u_j \in P \) could also compute \( CK’ = Q(h(I_j)) \oplus h(SK_{0j} \parallel I_0 \parallel I_j) \), but \( CK’ \neq CK \). Therefore, any \( u_j \in P \) cannot obtain the correct conference key.

4. SECURITY ANALYSES

The security of the self-certified key distribution and that of conference key distribution protocols are based on the well-known cryptographic assumption:

One-way hash function (OWHF): Given \( h(M) \), it is computationally infeasible to find \( M \), where \( h \) is a one-way hash function [3]. Moreover, it is computationally infeasible to find \( M \) and \( M’ \) such that \( h(M) = h(M’) \) [14].

Discrete logarithm problem (DLP): Given a prime \( p \), a generator \( \alpha \) of a large prime order \( q \) over GF\((p)\) where \( q \parallel p - 1 \), and \( y \in Z \), it is computationally infeasible to derive \( x \) such that \( y = \alpha^x \mod p \) [14]. Moreover, one can believe that it is computationally infeasible to find \( x \) such that \( y = x\alpha^x \mod p \) for given any \( y \in Z \) [3, 4].

We discuss the security of the private keys, shared keys, and conference keys used in the proposed self-certified key distribution and self-certified conference key distribution protocols as follows. We also demonstrate some possible but essential attacks against the proposed protocols and show that these attacks would not succeed in effect.

4.1. Compromise of Confidentiality of Private Keys

The private keys used in the proposed self-certified key distribution and conference key distribution protocols include the private key of the SA, the user’s chosen master key, and the user’s derived self-certified private key. Since the proposed protocols use the same key derivation, we discuss only the security of the private keys used in the self-certified key distribution protocols.
Suppose that an adversary attempts to compromise the SA’s private key from all available public or intercepted information. It is straightforward to see that the adversary will face the DLP to compute the SA’s private key $\gamma$ from the corresponding public key $\beta$ where $(\gamma, \beta)$ defined in the system setup stage. Moreover, the adversary can derive $\gamma$ from the intercepted $w_i$ unless he or she knows the time-variant parameter $z$. However, the time-variant variable $z$ is protected by the DLP.

Consider the scenario that an adversary attempts to reveal any registering user’s master key $k_i$ chosen in the user registration stage. One can see that computing $k_i$ from Eqn. (1) is based on the intractability of solving the DLP and the OWHF assumptions.

Suppose that an adversary attempts to derive the self-certified private key $x_i$ of the registering user $u_i$ from all available information. With the knowledge of the corresponding self-certified public key, it can be seen that computing $x_i$ from Eqn. (5) is based on the intractability of the DLP. Moreover, the adversary can reveal $x_i$ from the intercepted message $w_i$ as Eqn. (4) only if he or she knows the user’s chosen master key $k_i$ whose security is based on the DLP and the OWHF assumptions as discussed above.

4.2. Compromise of Authenticity of Private-Keys/Public-Keys

The authenticity requirement of private-keys/public-keys is to ensure that a private-key/public-key is associated with the true identity of the user. In this section we discuss some possible attacks on compromise of authenticity of private-keys/public-keys and show that these attacks would not succeed in effect. Consider the scenario of an active adversary $u_{\text{adv}}$ who attempts to choose a non-registering identity information $I_{\text{adv}}$ and contrive a valid self-certified private-key/public-key pair $(x_{\text{adv}}, y_{\text{adv}})$ satisfying Eqn. (5) without the assistance of the SA. From Eqn. (5), it can be seen that $u_{\text{adv}}$ cannot plot such an attack under the DLP and the OWHF assumptions.

Suppose that an active adversary may try to masquerade as some user $u_i$ to deceive $u_i$ into believing that he or she shares a secret key with $u_i$ associated with $I_i$ by plotting substitution or replaying attacks. This attack can be examined by a simple way such as [7], unless the adversary knows $x_i$ or finds another valid private-key/public-key pair $(x_i', y_i')$ associated with $I_i$. As discussed in Section 4.1. and previous paragraph, the adversary cannot mount this attack under the intractability of solving the DLP or/and reversing the adopted OWHF.

4.3. Deduction of the Shared Keys/Conference Keys

In the proposed key distribution and conference key distribution protocols, we consider a case that a passive adversary attempts to deduce a shared key/conference key to threaten the confidentiality using the eavesdropped
information. Some possible but essential attacks on deduction of the shared keys/conference keys are as follows.

Suppose that an adversary aside from specifically identified user attempts to gain access to a shared key in KDP1/KDP2. From Eqns. (6) or (7), one can see that the adversary can deduce the shared key from available intercepted information, unless he or she knows the private key protected by the intractability of the DLP as discussed in Section 4.1.

Consider a known-key attack that if compromise of past shared keys allows either a passive adversary to compromise future shared keys. The KDP1 is vulnerable to the known-key attacks, since the established shared key between two specified identified users is always fixed and the same. However, the KDP2 can withstand this attack because of the unknown time-variant variable $r_j$ protected by the DLP. In the proposed conference key distribution protocol, the adversary can compromise the conference key only if he or she knows the shared key between the chairman and any participant. Thus, the security of the proposed conference key distribution against this attack depends on the security of the adopted key distribution protocol.

Suppose that an adversary attempts to deduce the conference key established in the proposed conference key distribution protocol from available public or intercepted information. From the broadcasted authentic message polynomial $Q(x)$ as Eqn. (15), the adversary can obtain some information with the inputs of $h(I_j)$ or $h(SK_{oi} \parallel I_o \parallel I_j)$ where $u_j \in P$ and $u_j \in P$. In the case of the input of $h(I_j)$, the adversary cannot obtain the correct conference key $CK$ since $Q_1(h(I_j)) + cQ_2(h(I_j)) \neq CK \oplus h(SK_{oi} \parallel I_o \parallel I_j)$ as Eqn. (14). The probability that the adversary successfully finds the integer $c$ is $1/p$, which is negligible. In the case of the input of $h(SK_{oi} \parallel I_o \parallel I_j)$, the adversary can compromise $CK$ unless he knows $SK_{oi}$. Under intractability of solving the DLP, the adversary cannot obtain $SK_{oi}$ as discussed previously.

5. PERFORMANCE EVALUATION

We evaluate the performance of the proposed protocols in terms of the computational complexities, the communication costs. We denote some notations to facilitate the performance evaluation as follows. $T_{exp}$: the time for computing a modular exponentiation; $T_{mul}$: the time for computing a modular multiplication; $T_{inv}$: the time for computing a modular inverse; $|x|$: the size of an integer/set $x$.

The time for computing a modular addition/subtraction over $\text{GF}(p)$ or $\text{GF}(q)$, and that for computing an exclusive-or are ignored, since they are negligible as compared with those listed above. The computational complexities required in the proposed protocols are showed in Table 1. Moreover, Table 2 depicts the communication costs required in the proposed protocols. According to Table 1, it takes each user $3 \; T_{exp} + T_{mul} + 2T_h$ to register with SA, and it takes the SA $T_{exp} + 2T_{mul} + T_h$ to handle the registration information. The computational costs for each user in KDP1 and KPD2 are $2 \; T_{exp} + T_{mul} + T_h$ and $3 \; T_{exp} + 2T_{mul} + T_h$, respectively.

International Journal of Business and Information
respectively. In the conference key distribution protocol, it takes the chairman
\((2(\mid P \mid)^2 - \mid P \mid + \mid \bar{P} \mid)T_{mul} + (2 \mid \bar{G} \mid)T_h + (\mid P \mid)^2 - \mid P \mid)T_{inv}\) to initiate and
broadcast the authentic message, and the computational costs for each participant
to derive the shared conference key are \(\mid P \mid T_{mul} + 2T_h\). According the above
analysis, the proposed protocols run in polynomial time.

**Table 1**

**Computational Complexities for the Proposed Protocols**

<table>
<thead>
<tr>
<th>USER REGISTRATION</th>
<th>KEY DISTRIBUTION PROTOCOL</th>
<th>CONFERENCE KEY DISTRIBUTION PROTOCOL*2</th>
</tr>
</thead>
</table>
| Registering user:  | KDP1: \(2T_{exp} + T_{mul} + T_h\) | The chairman: 
\((2(\mid P \mid)^2 - \mid P \mid + \mid \bar{P} \mid)T_{mul} + (2 \mid \bar{G} \mid)T_h\)  
\(+ (\mid P \mid)^2 - \mid P \mid)T_{inv}\) 
Each participant: 
\(\mid P \mid T_{mul} + 2T_h\) |
| SA: \(T_{exp} + 2T_{mul} + T_h\) | KDP2: \(3\) \(T_{exp} + 2T_{mul} + T_h\) |

Remarks:
*1: The computational complexities include public key verification.
*2: Suppose that all shared keys are established by KDP1 or KDP2 in advance.

In the user registration stage, the required communication costs are
\(\mid I \mid + 2 \mid p \mid + \mid q \mid\). The communication costs in KDP1 and KDP2 are
\(2\mid I \mid + 2 \mid p \mid\) and \(2\mid I \mid + 4 \mid p \mid\), respectively. In the conference key distribution
protocol, the total communication costs are \(\mid I \mid + (\mid P \mid + 1) \mid p \mid\). The
communication costs in each stage variable depend on the length of identity
information, the length of large primes, and the total number of users.
Certificateless Authentic Key Distribution and Conference Key Distribution Schemes

Table 2
Communication Costs for the Proposed Protocols

<table>
<thead>
<tr>
<th>User Registration</th>
<th>Key Distribution Protocol</th>
<th>Conference Key Distribution Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>I</td>
<td>+2</td>
</tr>
<tr>
<td>(</td>
<td>I</td>
<td>+4</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

We proposed two authenticated key distribution protocols and one conference key protocol using identity-based, self-certified public keys. As discussed in our security analyses, the proposed protocols are secure against some possible but potential attacks in effect under the intractability of solving the DLP and reversing the adopted OWHF.

ACKNOWLEDGMENT

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 97-2745-P-001-001-, NSC 98-2219-E-011-001-, NSC 98-2221-E-011-073-MY3, and NSC 98-2218-E-011-018-.

REFERENCES

ABOUT THE AUTHORS

Chien-Lung Hsu is an associate professor in the Department and Graduate Institute of Information Management, Chang Gung University, Taiwan. He completed his Ph.D. in the Department of Information Management from National Taiwan University of Science and Technology, Taiwan, in 2002. His research interests include cryptography, information theory, and network security.
Tzong-Chen Wu is a professor in the Department of Information Management, National Taiwan University of Science and Technology, Taiwan. He completed his Ph.D. in the Department of Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 1992. Now, he serves as Direct Chair of the CCISA. He is also a member of IEEE and ACM. His current research interests include cryptography, data security, network security, and data engineering.

Kuo-Yu Tsai is a post-doctoral student at the Taiwan Information Security Center, National Taiwan University of Science and Technology, Taiwan. He received his Ph.D. in the Department of Information Management from National Taiwan University of Science and Technology, Taiwan, in 2009. His recent research interests include cryptography, wireless security, and network security.