INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). **DOI**: 10.1002/dac.2569

An authenticated group key distribution mechanism using theory of numbers

Yanjun Liu[1,2], Lein Harn[3] and Chin-Chen Chang[2,4,*,†]

[1]*School of Computer Science and Technology, Anhui University, Hefei, 230039, China*
[2]*Department of Computer Science and Information Engineering, Asia University, Taichung, 413, Taiwan*
[3]*Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, Kansas City, Missouri 64110-2499, USA*
[4]*Department of Information Engineering and Computer Science, Feng Chia University, Taichung, 407, Taiwan*

ABSTRACT

A group key distribution protocol can enable members of a group to share a secret group key and use it for secret communications. In 2010, Harn and Lin proposed an authenticated group key distribution protocol using polynomial-based secret sharing scheme. Recently, Guo and Chang proposed a similar protocol based on the generalized Chinese remainder theorem. In this paper, we point out that there are some security problems of Guo and Chang's protocol and propose a simpler authenticated group key distribution protocol based on the Chinese remainder theorem. The confidentiality of our proposed protocol is unconditionally secure. Copyright © 2013 John Wiley & Sons, Ltd.

Received 28 January 2013 Revised 29 March 2013 Accepted 15 April 2013

KEY WORDS: group key distribution; Chinese remainder theorem (CRT); secret sharing (SS); *t*-threshold range

1. INTRODUCTION

The group communication has been developed extensively in many applications currently. Ensuring the security of a group communication has become one of the most important issues in the development. Generally speaking, the security properties of a group communication include two basic aspects, that is, (i) the messages transmitted within the group can only be shared by authorized group members, but not by any unauthorized users; and (ii) the transmitted messages must be able to be authenticated by members. The security properties mentioned previously imply that a group session key must be used by authorized group members to encrypt and authenticate the messages.

Group key management [1] can be used for generating and distributing a group session key. Group key agreement protocols [2–6] and group key distribution protocols [7–11] are two kinds of group key management protocols. In key agreement protocols, all authorized group members work together to generate and distribute a group session key. Thus, there is no need to adopt a mutually trusted server. On the other hand, in group key distribution protocols, a mutually trusted server, usually called the *key generation center* (KGC), is employed to choose a group session key, and the KGC sends the session key to all authorized group members secretly. In general, group key distribution protocols are more efficient than key agreement protocols because in group key distribution protocols, a trusted KGC handles most key distribution tasks. A large amount of research works on this subject have been published in the literature [3–6].

*Correspondence to: Chin-Chen Chang, Department of Information Engineering and Computer Science, Feng Chia University, No. 100 Wenhwa Rd., Seatwen, Taichung, 407, Taiwan.
†E-mail: alan3c@gmail.com

In 2010, Harn and Lin [12] proposed an authenticated group key distribution protocol based on Shamir's (t, n) secret sharing (SS). In an (t, n) SS [13–16], a dealer divides a secret s into n shares. The two security requirements of an SS are (i) the secret s can be recovered by any t or more than t shares, and (ii) the secret s cannot be reconstructed by fewer than t shares. Shamir's (t, n) SS [13, 14], which is based on the Lagrange interpolating polynomial, is the most well-known SS scheme, and it has been studied extensively in the literature [17–20]. Shamir's (t, n) SS is unconditionally secure, that is, it can satisfy the two security requirements of an SS described previously without making any computational assumption. Aside from the polynomial-based SSs, SSs can also be based on the Chinese remainder theorem (CRT), such as Mignotte's SS scheme [19] and Asmuth–Bloom's SS scheme [20].

Recently, Guo and Chang [21] pointed out that although Harn and Lin's protocol can withstand outside and inside attacks, a random challenge, R_i, which is a component of the group key's construction, must be transmitted from each authorized group member to the KGC. Guo and Chang proposed a new authenticated group key distribution protocol based on the concepts of the generalized Chinese remainder theorem (GCRT) [22, 23]. They claimed that by using the GCRT, their protocol can avoid sending random challenges.

Although (t, n) SS is an important tool in designing group key distribution protocols, (t, n) SS can be widely used for applications in other cryptographic topics. Parakh and Kak [24] proposed a space efficient SS scheme, in which k secrets can be divided into n shares, where $k \leq n$. Their method can be used for secure parallel communication and online data storage. Chen and Wu [25] presented an anonymous multipath routing protocol based on SS in mobile ad hoc networks. The protocol can provide high security by preventing passive attacks and reduce the successful probability of active attacks. Guo and Chang [26] proposed an SS scheme for general access structures such that the shared secret only can be recovered by any qualified subsets of participants. Zhu *et al*. [27] presented an N-party cloud storage protocol based on SS, which achieves authentication, confidentiality, and entangled security. In 2012, Harn [28] proposed a novel method of group authentication to authenticate all users belonging to the same group. The method is based on the SS and is a many-to-many type of authentication, which involves multiple provers and multiple verifiers. Unlike Harn's paper that is focused on the authentication issue for group-oriented applications, our paper will discuss how to securely distribute a group key to members in group communications.

Inspired by Harn and Lin's protocol [12] and Guo and Chang's protocol [21], in this paper, we propose a simple authenticated group key distribution protocol based on the CRT. The contributions of our protocol are listed as follows:

(1) We point out some security problems of Guo and Chang's protocol.
(2) Our proposed protocol based on the CRT is much simpler than the other two protocols.
(3) In our proposed protocol, the group key confidentiality is unconditionally secure, whereas the security of Harn and Lin's protocol relies on the RSA assumption [12].

The rest of this paper is organized as follows. In Section 2, we provide some preliminaries. Our proposed protocol is described in Section 3. Section 4 offers analysis of our protocol. Conclusion is given in Section 5.

2. PRELIMINARIES

In this section, we briefly introduce some fundamentals that are essential in the design of our protocol. First, we review the CRT and then describe two CRT-based SSs, that is, Mignotte's (t, n) SS [19] and Asmuth–Bloom's (t, n) SS [20]. At last, we review Guo and Chang's group key distribution protocol [21] that uses Asmuth–Bloom's (t, n) SS and the GCRT [22, 23] as its building blocks.

2.1. Chinese remainder theorem

The CRT [22] can be described as follows. Given t pairwise, co-prime integers, $p_1, p_2, \ldots, p_t$, with $\mathrm{GCD}(p_i, p_j) = 1$ for $i \neq j$,the following system of simultaneous congruencies

DOI: 10.1002/dac

$$\begin{aligned} X &= x_1(\bmod p_1),\\ X &= x_2(\bmod p_2),\\ &\vdots\\ X &= x_t(\bmod p_t), \end{aligned}$$

has one unique solution X in Z_P, where $P = \prod_{i=1}^{t} p_i$ and Z_P stands for integers in the range of $[0,P)$. From CRT, X can be computed as $X = \sum_{i=1}^{t} M_i \cdot M'_i \cdot x_i (\bmod P)$, where $M_i = \frac{P}{p_i}$ and $M_i \cdot M'_i \equiv 1 (\bmod p_i)$.

2.2. *Mignotte's (t, n) SS*

Mignotte's (t, n) SS [19] was introduced in 1983, which is based on the CRT. It consists of two phases, that is, share generation and secret reconstruction.

Share generation

A set of n positive integers, $\{p_1 < p_2 < \ldots < p_n\}$, is selected satisfying the following conditions:

(1) $\mathrm{GCD}(p_i, p_j) = 1$ for $i \neq j$,
(2) $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$,

where p_i is the public information assigned to shareholder, u_i. Now let us define the *t-threshold range*. The t-threshold range is denoted as $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$, which represents integers in the range of $(p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t)$.
The dealer selects the secret s in the t-threshold range, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$. Then, the share s_i of shareholder, u_i, is computed as $s_i = s (\bmod p_i)$, $i = 1, 2, \ldots, n$. Each share, s_i, is sent to shareholder, u_i, in a secure channel.

Secret reconstruction

Given any t distinct shares, for example, $s_{lj} \in \{s_1, s_2, \ldots, s_n\}$ for $j = 1, 2, \ldots, t$, the secret s can be recovered by constructing the following system of simultaneous congruencies:

$$\begin{aligned} s &= s_{l1}(\bmod p_1),\\ s &= s_{l2}(\bmod p_2),\\ &\vdots\\ s &= s_{lt}(\bmod p_t). \end{aligned}$$

According to the CRT, the unique solution s in $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$ can be computed as

$$s = \sum_{j=1}^{t} M_j \cdot M'_j \cdot s_{lj} (\bmod P), \text{ where } P = \prod_{j=1}^{t} p_j,\ M_j = \frac{P}{p_j} \text{ and } M_j \cdot M'_j \equiv 1 (\bmod p_j).$$

There is one important reason why the secret s must be selected in the t-threshold range, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$. This can ensure that Mignotte's (t, n) SS satisfies two security requirements of an SS. That are (i) the secret s can be recovered successfully by any t or more than t shares, and (ii) the secret cannot be recovered by fewer than t shares. The lower bound, $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n$, of the t-threshold range, is the largest product of any $t-1$ moduli, and the upper bound, $p_1 \cdot p_2 \cdot \ldots \cdot p_t$, is the smallest product of any t moduli of the set of n positive integers, $\{p_1, p_2, \ldots, p_n\}$. If the secret s is chosen in the t-threshold range, the product of any t moduli associated with t shares is surely larger than or equal to the upper bound, $p_1 \cdot p_2 \cdot \ldots \cdot p_t$, which guarantees that the secret s can be recovered by any t or more than t shares, and the product of any $t-1$ moduli associated with $t-1$ shares is smaller than or equal to the lower bound, $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n$, which ensures that the secret s cannot be reconstructed by fewer than t shares. However, the weakness of Mignotte's (t, n) SS

DOI: 10.1002/dac

is that some information of the secret s can be revealed with knowing fewer than t shares, which implies that it is not a perfect SS.

2.3. Asmuth–Bloom's (t, n) SS

In 1983, Asmuth and Bloom presented an (t, n) SS based on the CRT [20]. It consists of two phases as follows:

Share generation
A set of positive integers, $\{q, p_1 < p_2 < \ldots < p_n\}$, is selected satisfying the following conditions:

(1) $\mathrm{GCD}(p_i, p_j) = 1$ for $i \neq j$,
(2) $\mathrm{GCD}(q, p_i) = 1$ for all i,
(3) $q \cdot p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$,

where p_i is the public information assigned to shareholder, u_i. The dealer selects the secret s in Z_q. Then, the dealer computes an integer $X = s + dq$, in the t-threshold range, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$, where d is an integer. The share s_i of shareholder, u_i, is generated by computing $s_i = X (\mathrm{mod}\, p_i)$, $i = 1, 2, \ldots, n$. Each share, s_i, is sent to shareholder, u_i, in a secure channel.

Secret reconstruction
Given any t distinct shares, for example, $s_{lj} \in \{s_1, s_2, \ldots, s_n\}$ for $j = 1, 2, \ldots, t$, the integer X can be recovered by constructing the following system of simultaneous congruencies:

$$\begin{aligned} X &= s_{l1} (\mathrm{mod}\, p_1), \\ X &= s_{l2} (\mathrm{mod}\, p_2), \\ &\vdots \\ X &= s_{lt} (\mathrm{mod}\, p_t). \end{aligned}$$

From CRT, the unique solution X in $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$ can be computed as $X = \sum_{j=1}^{t} M_j \cdot M'_j \cdot s_{lj} (\mathrm{mod}\, P)$, where $P = \prod_{j=1}^{t} p_j, M_j = \frac{P}{p_j}$ and $M_j \cdot M'_j \equiv 1 (\mathrm{mod}\, p_j)$. Once X is determined, the secret s can be recovered by computing $s = X (\mathrm{mod}\, q)$.

In Asmuth–Bloom's (t, n) SS, both security requirements of an SS can be satisfied by choosing the integer X in the t-threshold range, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t}$. Additionally, Asmuth–Bloom's (t, n) SS is a perfect SS because no information of the secret can be disclosed with knowing fewer than t shares. Interest reader can refer to the original paper [20] for detailed discussion.

2.4. Guo and Chang's group key distribution protocol

In this subsection, we review Guo and Chang's authenticated group key distribution protocol [21], which is based on the GCRT.

Guo and Chang's protocol consists of three phases, that is, the initialization phase, registration phase, and group key distribution phase. Suppose that t group members want to hold a conference. In the initialization phase, the KGC generates an Asmuth–Bloom sequence, $\{q, p_1 < p_2 < \ldots < p_t\}$. In the registration phase, the KGC randomly selects a group session key K in Z_q and then generates an integer $X = K + dq$, where d is an integer and X is in $Z_{p_1 \cdot p_2 \cdot \ldots \cdot p_t}$. After that, shares, x_i, for $i = 1, 2, \ldots, t$, of X are generated by the GCRT. (x_i, p_i) is the secret (i.e., also called *private share*) between the KGC and each authorized group member u_i. In the group key distribution phase, the KGC extends the Asmuth–Bloom sequence, $\{q, p_1 < p_2 < \ldots < p_t\}$, to a longer sequence, $\{q, p_1 < p_2 < \ldots < p_t < p_{t+1} < \ldots < p_{2t-1}\}$ and then uses the new sequence and

DOI: 10.1002/dac

the GCRT to generate $(t-1)$ public shares. Thus, each u_i can recover the group key K by using his/her private share, (x_i,p_i), and $(t-1)$ public shares.

Unfortunately, Guo and Chang's protocol has the following security problems. (i) Because the integer X is in $Z_{p_1 \cdot p_2 \cdot \ldots \cdot p_t}$, not in the t-threshold range, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n \cdot p_1 \cdot p_2 \cdot \ldots \cdot p_t}$, this may violate one of the security requirements of an SS, that is, the integer X may be obtained by fewer than t shares, and (ii) because the public moduli are all larger than private moduli, it is possible to recover the integer X by $(t-1)$ public shares. We will develop a simpler authenticated group key distribution protocol based on the CRT in the next section.

3. OUR PROPOSED PROTOCOL

In this section, we propose a simple authenticated group key distribution protocol, which is based on the CRT. The KGC first establishes a secret with each authorized group member in the registration. Then, in real-time operation, the KGC determines a group session key based on shared secrets with all group members via the CRT and transmits the group key to all group members by broadcasting public shares. Each group member can use his/her secret shared with the KGC as the private share and the public shares to recover the group key.

Our protocol consists of three phases: (i) the registration, (ii) the generation of the group key, and (iii) the distribution of the group key. Details of our protocol are included in the following.

3.1. Registration

Each user in a communication needs to register at the KGC initially. Let us assume that there are n users, u_i, for $i = 1, 2, \ldots, n$. The KGC randomly selects n positive, pairwise, co-prime integers, $p_1, p_2, \ldots, p_n$, and then chooses n positive integers, $x_1, x_2, \ldots, x_n$, satisfying $x_i < p_i$ for $i = 1, 2, \ldots, n$. The KGC shares the secret (x_i, p_i) (also called private share) with each user u_i in a secure channel.

Registration phase for group members is shown in Figure 1.

3.2. Generation of the group key

Let $U = \{u_1, u_2, \ldots, u_t\}$ denote a group consisting of t users who attempt to have a group communication. As illustrated in Figure 2, the KGC generates a group session key K following the steps shown as follows:

Step (1) The initiator (a user) delivers a key generation request of a group communication involving group members, $U = \{u_1, u_2, \ldots, u_t\}$, to the KGC.

Step (2) The KGC broadcasts a random integer, R, satisfying $R < p_j$ for $j = 1, 2, \ldots, t$, to all group members.

Step (3) Each group member, u_i, sends a random integer, R_i, satisfying $R_i < p_j$ for $j = 1, 2, \ldots, t$, to the KGC.

Step (4) With the secrets shared with all group members and random integers, the KGC establishes a system of simultaneous congruencies:

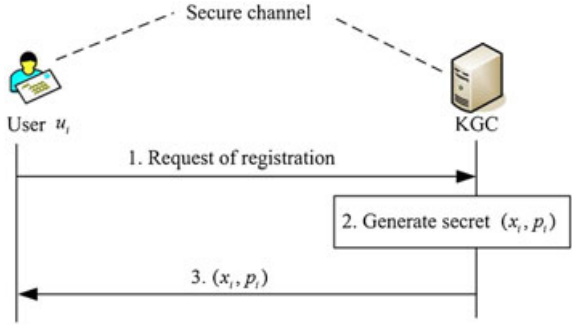

Figure 1. Registration phase.

DOI: 10.1002/dac

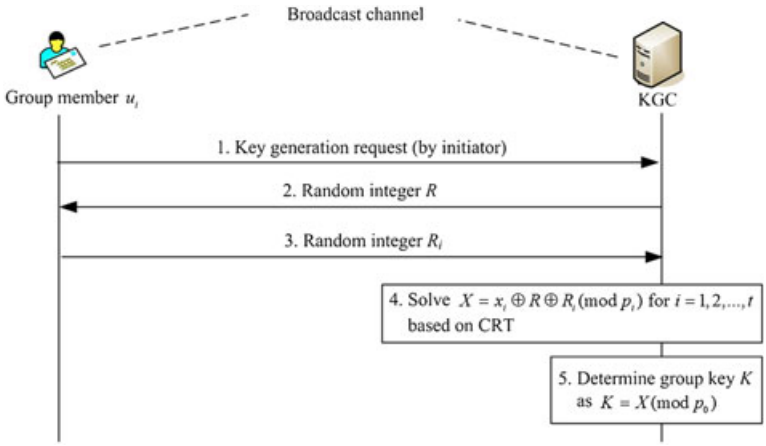

Figure 2. Group key generation phase.

$$
\begin{aligned}
X &= x_1 \oplus R \oplus R_1 (\bmod p_1),\\
X &= x_2 \oplus R \oplus R_2 (\bmod p_2),\\
&\vdots\\
X &= x_t \oplus R \oplus R_t (\bmod p_t).
\end{aligned}
$$

From the CRT, the KGC computes the unique solution X in Z_P, where $P = \prod_{i=1}^{t} p_i$.

Step (5) The group key K is determined as $K + dp_0 = X$ such that $0 \le K < p_0$, where d is an integer and $p_0 < \min\{p_1, p_2, \ldots, p_t\}$.

3.3. Distribution of the group key

During this phase, the KGC distributes the group key K by broadcasting information as public shares to all authorized group members. As a result, each group member can use his/her secret (private share) shared with the KGC and the public information (public shares) to recover the group key. Furthermore, mutual key confirmation can be achieved. Each group member can verify whether the group key he/she obtained is valid by checking a separate authentication message transmitted from the KGC; on the other hand, the KGC can check whether the group key that he/she generated is identical to the one that each group member has sent to him/her. The phase of the group key distribution consists of following six steps. It is shown in Figure 3.

Step (1) The KGC picks $(t-1)$ positive integers, $p_1', p_2', \ldots, p_{t-1}'$, which satisfy the following conditions:

(1) $\mathrm{GCD}(p'_i, p'_j) = 1$ for $i = 1, 2, \ldots, t-1, j = 1, 2, \ldots, t-1$ and $i \neq j$,

(2) $\mathrm{GCD}(p'_i, p_k) = 1$ for $i = 1, 2, \ldots, t-1$ and $k = 1, 2, \ldots, t$,

(3) X is in the t-threshold range, $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}, p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$, where $p'' = \min\{p_i, i = 1, 2, \ldots, t\}$.

Step (2) The KGC computes $x'_i = X (\bmod p'_i)$ for $i = 1, 2, \ldots, t-1$. The KGC also generates an authentication message, $Auth_1 = h_1(K, p_0, R, R_1, R_2, \ldots, R_t, (x_1', p_1'), (x_2', p_2'), \ldots, (x_{t-1}', p_{t-1}'))$, where h_1 is a collision-free, one-way hash function.

Step (3) The KGC broadcasts p_0, $\{(x_i', p_i')\}_{1 \le i \le t-1}$, and $Auth_1$ to all authorized group members.

Step (4) Each authorized group member u_i uses his/her private share, $(x_i \oplus R \oplus R_i, p_i)$, and $(t-1)$ public shares, $\{(x_i', p_i')\}_{1 \le i \le t-1}$, to construct a system of simultaneous congruencies:

DOI: 10.1002/dac

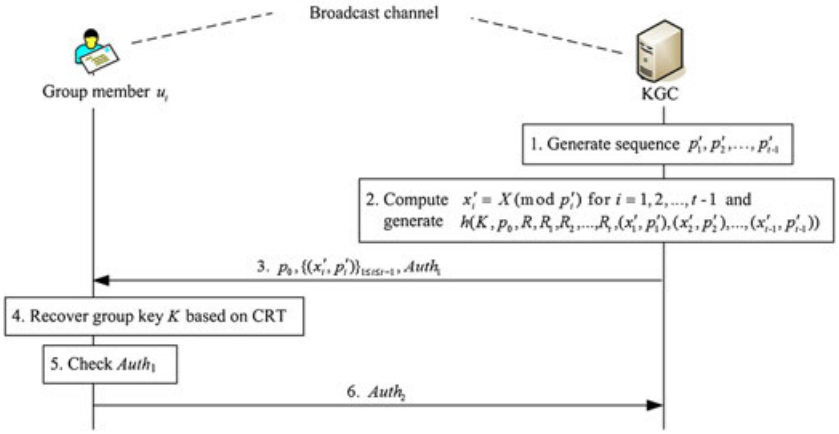

Figure 3. Group key distribution phase.

$$\begin{cases} X = x_i \oplus R \oplus R_i \pmod{p_i}, \\ X = x'_i \pmod{p'_i}, i = 1, 2, \ldots, t\text{-}1. \end{cases}$$

The unique solution X in $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$ can be computed from the CRT. Thus, each u_i can recover the group key as $K = X \pmod{p_0}$.

Step (5) Each authorized group member u_i computes $h_1(K, p_0, R, R_1, R_2, \ldots, R_t, (x'_1, p'_1), (x'_2, p'_2), \ldots, (x'_{t-1}, p'_{t-1}))$, and checks whether the value is identical to $Auth_1$. If the two values are identical, u_i authenticates that the group key K was sent by the KGC.

Step (6) u_i sends $Auth_2 = h_2(K, p_0, R)$ to the KGC, where h_2 is a collision-free, one-way hash function. The KGC computes $h_2(K, p_0, R)$ and checks whether the value is identical to $Auth_2$. If the two values are identical, mutual key confirmation is achieved.

4. ANALYSIS

In this section, we analyze the protocol and show that our proposed group key distribution protocol possesses the following properties:

- Providing key confidentiality
- Providing key authentication
- Providing one-time group key
- Providing reuse of secrets
- Providing key updating
- Providing forward and backward secrecy

4.1. Key confidentiality

Our proposed protocol can provide key confidentiality due to the property of the SS scheme. In our protocol, the KGC first generates an integer X via the CRT based on t private shares, $(x_i \oplus R \oplus R_i, p_i)$, for $i = 1, 2, \ldots, t$. Then, the KGC selects $(t-1)$ positive, pairwise, co-prime integers, $p'_1, p'_2, \ldots, p'_{t-1}$, such that X is in the t-threshold range, $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$, where $p'' = \min\{p_i, i = 1, 2, \ldots, t\}$. The KGC broadcasts $(t-1)$ public shares, (x'_i, p'_i), for $i = 1, 2, \ldots, t-1$ to all authorized group members. Upon receiving public shares, each authorized group member can use a total of t shares, that is, his/her private share, $(x_i \oplus R \oplus R_i, p_i)$, and $(t-1)$ public shares, $\{(x'_i, p'_i)\}_{1 \le i \le t-1}$, to compute X,

DOI: 10.1002/dac

and then to recover the group key K, but illegal users cannot obtain any useful information about the group key.

Any user who does not belong to the group and makes effort to obtain the group key shared among all authorized group members is called an *outside attacker*. The outside attacker can only receive $(t-1)$ public shares, $\{(x'_i,p'_i)\}_{1 \le i \le t-1}$, broadcasted by the KGC and uses these public shares to construct a system of simultaneous congruencies as follows:

$$\begin{aligned} X' &= x'_1 \left(\bmod p'_1\right), \\ X' &= x'_2 \left(\bmod p'_2\right), \\ &\vdots \\ X' &= x'_{t-1} \left(\bmod p'_{t-1}\right). \end{aligned}$$

From the CRT, the outside attacker computes the integer X' as $X' = \sum_{i=1}^{t-1} N_i \cdot N'_i \cdot x'_i \left(\bmod P'\right)$, where $P' = \prod_{i=1}^{t-1} p'_i$, $N_i = \frac{P'}{p'_i}$ and $N_i \cdot N'_i \equiv 1 (\bmod p'_i)$. However, the integer X' is in $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}}$, which is different from X in the t-threshold range, $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}, p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$, where $p'' = \min\{p_i, i = 1, 2, \ldots, t\}$. Therefore, an outside attacker cannot obtain the real X.

Next, we analyze whether the group key K can be reconstructed by an outside attacker from X'. Although X' is different from X, there exists one relationship between these two values, that is, $X = X' + \beta p'_1 p'_2 \ldots p'_{t-1}$, where β is an integer. If the value, β, which shifts $X' \in Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}}$ to $X' + \beta p'_1 p'_2 \ldots p'_{t-1} \in Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}, p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$, can be determined, the value, X, can be obtained. However, there are more than p_0 possible values of β (i.e., $\frac{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}} > p_0$), which can shift X' into the t-threshold range, $Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1}, p'_1 \cdot p'_2 \cdot \ldots \cdot p'_{t-1} \cdot p''}$. But, there is only one value of β, which can shift X' to the value X. The successful probability of this approach is smaller than the probability of randomly guessing the secret group key K. Therefore, no information is disclosed from these $(t-1)$ public shares. The security of this attack is perfect secrecy. This implies that our proposed protocol can provide group key confidentiality. Moreover, the group key confidentiality is unconditionally secure because it does not depend on any computational assumptions.

4.2. Key authentication

To provide group key authentication, the KGC broadcasts an authentication message, $Auth_1 = h_1(K, p_0, R, R_1, R_2, \ldots, R_t, (x'_1,p'_1), (x'_2,p'_2), \ldots, (x'_{t-1},p'_{t-1}))$, to all authorized group members in the group key distribution, where h_1 is a collision-free, one-way hash function. The authentication message $Auth_1$ uses the group key K, public information, p_0, the transmitted random integers, $R, R_1, R_2, \ldots, R_t$, and public pairs $(x'_1,p'_1), (x'_2,p'_2), \ldots, (x'_{t-1},p'_{t-1})$, as its inputs. As discussed in subsection 4.1, only authorized group members can obtain the group key K. An outside attacker cannot forge an authentication message because he/she does not have a valid secret shared with the KGC. In addition, any group member cannot forge an authentication message because he/she does not know other authorized members' secrets. Therefore, each authorized group member can authenticate that the group key is indeed transmitted by the KGC rather than by an attacker. On the other hand, each authorized group member u_i sends $Auth_2 = h_2(K,p_0,R)$ to the KGC, where h_2 is another collision-free, one-way hash function. This can ensure mutual key confirmation.

4.3. One time group key

One time group key means that different group keys must be used for different communications involving the same group members. By providing one time group key, an outside attacker cannot

DOI: 10.1002/dac

reuse a previously compromised group key by replaying this compromised group key in future communications. In our proposed protocol, the group key is determined on the basis of t secrets, $(x_i \oplus R \oplus R_i, p_i)$, for $i = 1, 2, \ldots, t$, of group members, where R_i is an random integer of member, u_i. Therefore, when same group members starts a new conference, each member will selects a new random integer, R_i. This can ensure the property of one time group key.

4.4. Reuse of secrets

Let us consider a group member who attempts to obtain other authorized members' secrets shared with the KGC. This type of group member is also called an *inside attacker*. Because an inside attacker is an authorized member who has already known the group key, our proposed protocol needs to prevent the inside attacker from obtaining other authorized members' secrets. Because each authorized group member u_i's secret, (x_i,p_i), is sent from the KGC in a secure channel during the registration phase; it is impossible for the inside attacker to obtain other authorized group members' secrets. Thus, the secret shared with the KGC in the registration can be reused for multiple group key distributions. It is unconditionally secure, whereas the security of Harn and Lin's protocol relies on the RSA assumption [12].

4.5. Key updating

In our protocol, an integer X is determined on the basis of t secrets, $(x_i \oplus R \oplus R_i, p_i)$, for $i = 1, 2, \ldots, t$, of group members. The group key K is then computed as $K = X (\mathrm{mod}\, p_0)$. By adding/removing any member to/from the group, the KGC should update the group key to ensure that a previously used group key will not be used in the future group communication. In the following, we will describe the group key updating process, which can be implemented easily.

Suppose that a group consists of t members and each group member u_i shares a secret, (x_i,p_i), with the KGC. When a user u_j joins the group, the KGC randomly selects a pair of secrets, (x_j,p_j) that satisfy $\mathrm{GCD}(p_j,p_i) = 1$ for $i = 1, 2, \ldots, t$ and $x_j < p_j$, and shares them with u_j in a secure channel. Then, the KGC sends the random integer, R, to u_j and u_j sends a random integer, R_j, to the KGC. Afterward, the KGC computes an integer X'' by using $(t+1)$ secrets, $(x_i \oplus R \oplus R_i, p_i)$ for $i = 1, 2, \ldots, t+1$, via the CRT and updates the group key by $K' = X'' (\mathrm{mod}\, p_0)$. In the phase of key distribution, the KGC reselects t positive, pairwise, co-prime integers, $p'_1, p'_2, \ldots, p'_t$, such that $X'' \in Z_{p'_1 \cdot p'_2 \cdot \ldots \cdot p'_t, p'_1 \cdot p'_2 \cdot \ldots \cdot p'_t, p''}$, where $p'' = \min\{p_i, i = 1, 2, \ldots, t+1\}$. The KGC makes t pairs, $\{(x'_i,p'_i)\}_{1 \le i \le t}$, public known as public shares. Each authorized member in the new group can use his/her private share, $(x_i \oplus R \oplus R_i, p_i)$, and t public shares, $\{(x'_i,p'_i)\}_{1 \le i \le t}$, to compute the new integer X'' and recover the new group key K'.

When a member u_j leaves the group, the KGC deletes u_j's secret shared with the KGC. Thus, u_j becomes an unauthorized member. The KGC needs to update the group key for the $(t-1)$ members who remain in the group. The KGC uses $(t-1)$ secrets, $(x_i \oplus R \oplus R_i, p_i)$, for $i = 1, 2, \ldots, t-1$ to generate the new group key K' according to the CRT. Then, the KGC broadcasts $(t-2)$ public shares $\{(x'_i,p'_i)\}_{1 \le i \le t-2}$ to $(t-1)$ group members. As a result, each authorized member who remains in the group can use his/her private share, $(x_i \oplus R \oplus R_i, p_i)$, and $(t-2)$ public shares, $\{(x'_i,p'_i)\}_{1 \le i \le t-2}$, to reconstruct the new group key K'.

4.6. Forward and backward secrecy

Forward secrecy guarantees that it is unable to discover any previous group key by compromising the group's current key. As discussed in subsection 4.5, the KGC can update the group key K to K' when adding a member u_i to the group. By the key updating approach adopted in subsection 4.5, the new member u_i cannot use the current key K' and public shares to obtain the previous key K.

Backward secrecy ensures that any current group key cannot be determined from a previous key that has been compromised. As discussed in subsection 4.5, the KGC can update the group key K to K' when removing a member u_j from the group. By the key updating approach adopted in subsection 4.5, u_j cannot obtain the current key K' by the previous key K because he/she does not own a private share.

DOI: 10.1002/dac

5. CONCLUSION

In this paper, we proposed an authenticated group key distribution protocol based on the CRT. Each user needs to register and obtain a secret from the KGC initially. In real-time operation, the KGC can broadcast a secret group key to all members based on all members' secrets. The secret shared between each user and the KGC can be reused for multiple group communications. The confidentiality of our proposed protocol is unconditionally secure.

ACKNOWLEDGEMENTS

This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

REFERENCES

1. Boyd C. On key agreement and conference key agreement. *Proceedings of Second Australasian Conference on Information Security and Privacy*, Sydney, Australia, Jul. 1997; 294–302.
2. He DB, Chen JH, Hu J. A pairing-free certificateless authenticated key agreement protocol. *International Journal of Communication Systems* 2012; **25**(2):221–230.
3. Katz J, Yung M. Scalable protocols for authenticated group key exchange. *Journal of Cryptology* 2007; **20**:85–113.
4. Bohli JM. A framework for robust group key agreement. *Proceedings of International Conference on Computational Science and Applications* (*ICCSA'06*), Glasgow, UK, May 2006; 355–364.
5. Xie Q. A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems* 2012; **25**(1):47–54.
6. Lee PPC, Lui JCS, Yau DKY. Distributed collaborative key agreement and authentication protocols for dynamic peer groups. *IEEE/ACM Transactions on Networking* 2006; **14**(2):263–276.
7. Eltoweissy M, Heydari MH, Morales L, Sudborough IH. Combinatorial optimization of group key management. *Journal of Network and Systems Management* 2004; **12**(1):33–50.
8. Sherman AT, McGrew DA. Key establishment in large dynamic groups using one-way function trees. *IEEE Transactions on Software Engineering* 2003; **29**(5):444–458.
9. Perrig A, Song D, Tygar JD. Elk, a new protocol for efficient large group key distribution. *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, USA, May 2001; 247–262.
10. Chang CC, Wu TC, Chen CP. The design of a conference key distribution system. *Proceedings of Advances in Cryptology-AUSCRYPT'92*, Gold Coast, Australia, LNCS 0718, Dec. 1992; 459–466.
11. Steiner M, Tsudik G, Waidner M. Diffie-Hellman key distribution extended to group communication. *Proceedings of Third ACM Conference on Computer and Communication Security* (*CCS'96*), New Delhi, India, Mar. 1996; 31–37.
12. Shamir A. How to share a secret. *Communications of the ACM* 1979; **22**(11):612–613.
13. Blakley GR. Safeguarding cryptographic keys. *Proceedings of American Federation of Information Processing Societies National Computer Conference*, New York, USA, Nov. 1979; **48**:313–317.
14. Berkovits S. How to broadcast a secret. *Proceedings of Eurocrypt '91 Workshop Advances in Cryptology*, Brighton, UK, Apr. 1991; 547:536–541.
15. Laih C, Lee J, Harn L. A new threshold scheme and its application in designing the conference key distribution cryptosystem. *Information Processing Letters* 1989; **32**:95–99.
16. Li CH, Pieprzyk J. Conference key agreement from secret sharing. *Proceedings of the Fourth Australasian Conference Information Security and Privacy* (*ACISP '99*), Wollongong, Australia, Apr. 1999; 64–76.
17. Saze G. Generation of key predistribution schemes using secret sharing schemes. *Discrete Applied Mathematics* 2003; **128**:239–249.
18. Mignotte M. How to share a secret. *Proceedings of the Workshop on Cryptography, Lecture Notes in Computer Science* 1983; **149**:371–375.
19. Asmuth C, Bloom J. A modular approach to key safeguarding. *IEEE Transactions on Information Theory* 1983; Vol. **IT-29**(2):208–210.
20. Harn L, Lin CL. Authenticated group key transfer protocol based on secret sharing. *IEEE Transactions on Computers* 2010; **59**(6):842–846.
21. Guo C, Chang CC. An authenticated group key distribution protocol based on the generalized Chinese remainder theorem. *International Journal of Communication Systems* 2012; article in press.
22. Lai YP, Chang CC. Parallel computational algorithms for generalized Chinese remainder theorem. *Computers and Electrical Engineering* 2003; **29**(8):801–811.
23. Chang CC, Lee HC. A new generalized group-oriented cryptoscheme without trusted centers. *IEEE Journal on Selected Areas in Communications* 1993; **11**(5):725–729.

DOI: 10.1002/dac

24. Parakh A, Kak S. Space efficient secret sharing for implicit data security. *Information Sciences* 2011; **181**(2): 335–341.
25. Chen S, Wu M. Anonymous multipath routing protocol based on secret sharing in mobile ad hoc networks. *Journal of Systems Engineering and Electronics* 2011; **22**(3):519–527.
26. Guo C, Chang CC. A Construction for secret sharing scheme with general access structure. *Journal of Information Hiding and Multimedia Signal Processing* 2013; **4**(1):1–8.
27. Zhu H, Liu T, Zhu D, Li H. Robust and simple N-party entangled authentication cloud storage protocol based on secret sharing scheme. *Journal of Information Hiding and Multimedia Signal Processing* 2013; **4**(2):110–117.
28. Harn L. Group authentication. *IEEE Transactions on Computers* 2012; article in press.

AUTHORS' BIOGRAPHIES

Yanjun Liu received her B.S. degree in 2005, in School of Computer Science and Technology from Anhui University, Hefei, China. She received her Ph.D. degree in 2010, in School of Computer Science and Technology from University of Science and Technology of China, Hefei, China. She is currently serving in Anhui University. Meanwhile, she is a postdoc at Asia University, Taichung, Taiwan. Her current research interests include information security and computer cryptography.

Lein Harn received his B.S. degree in Electrical Engineering from the National Taiwan University in 1977. In 1980, he received his M.S. degree in Electrical Engineering from the State University of New York, Stony Brook, and in 1984, he received his Ph.D. degree in Electrical Engineering from the University of Minnesota. Currently, he is a Full Professor at the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, USA. His research interests are cryptography, network security, and wireless communication security. He has published number of papers on digital signature design and applications, wireless and network security.

Chin-Chen Chang received his Ph.D. degree in Computer Engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in Computer and Decision Sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. Prior to joining Feng Chia University, Professor Chang was an Associate Professor in Chiao Tung University, Professor in National Chung Hsing University, Chair Professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.

DOI: 10.1002/dac