A novel Group Key Transfer Protocol

Chingfang Hsu[1] Bing Zeng[1] Qi Cheng[2] Guohua Cui[1]

[1]Information Security Lab, College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan, China

[2]Engineering Department, Institute of Wuhan Digital Engineering, Wuhan, China

Abstract: Group key transfer protocols depend on a mutually trusted key generation center (KGC) to transport the group key to all group members secretly. This approach requires that a trusted sever be set up, and it incurs communication overhead costs. In addition, the existing group key transfer protocols based on secret sharing all use threshold schemes that need to compute a t-degree interpolating polynomial to encrypt and decrypt the secret group key, then it increases the computational complexity of system. In this paper, we first present a novel group key transfer protocol without an online KGC, which is based on DH key agreement and a perfect linear secret sharing scheme (LSSS). The confidentiality of the group key transfer phase of this protocol is information theoretically secure, which is ensured by this LSSS. Furthermore, this protocol can resist potential attacks and also reduce the overhead of system implementation. Goals and security threats of our proposed group key transfer protocol will be analyzed in detail.

Keywords: Group key transfer protocol, group key, DH key agreement, linear secret sharing schemes, monotone span programs, confidentiality.

1 Introduction

In order to ensure secure communication, before exchanging communication messages, a key establishment protocol will distribute one-time secret session keys to all participants, which needs to provide confidentiality and authentication for session keys. Namely, confidentiality ensures the sender that the message can be read only by an intended receiver and authentication ensures the receiver that the message was sent by a specified sender and

the message was not altered en route.

There are two types of key establishment protocols: key transfer protocols and key agreement protocols[5]. Key transfer protocols depend on a mutually trusted key generation center (KGC) to select session keys and then transport session keys to all communication users secretly. Generally, KGC encrypts session keys under another secret key shared with each user during registration. In key agreement protocols, all communication users are involved to determine session keys. The most commonly used key agreement protocol is Diffie-Hellman (DH) key agreement protocol [13]. However, DH public key distribution algorithm can only provide session key for two users; not for a group more than two members.

When a secure communication involves more than two users, a group key is needed for all group members. Most well-known group key management protocols can be classified into two categories: centralized group key management protocols and distributed group key management protocols[18].

The class of centralized group key management protocols is the most widely used group key management protocols. Harney et al. [16] proposed a group key management protocol that requires $O(n)$ encryptions to update a group key, where n is the size of group. A set of scalable hierarchical structure-based group key protocols [11], [27], [30] have been proposed. Fiat and Naor [15] proposed a k-resistant protocol, i.e., coalitions of up to k users are secure. Eltoweissy et al. [14] proposed a protocol based on Exclusion Basis Systems (EBS), a combinatorial formulation of the group key management problem.

Most distributed group key management protocols took natural generalization of the DH key agreement protocol, such as, Ingemarsson et al. [21], Steer et al. [31], Burmester and Desmedt [9], and Steiner et al. [32] followed this approach. In 1996, Steiner et al. proposed a natural extension of DH [32] and later in 2001, it has been enhanced with authentication services and has proved to be secure [6]. In 2006, Bohli [8] developed a framework for robust group key agreement. Then, in 2007, Bresson et al. [7] constructed a generic authenticated group DH Key exchange. Also, in 2007, Katz and Yung [22] proposed the first constant-round and fully scalable group DH protocol. The main feature of the group DH key exchange is to establish a secret group key among all group members without depending on a

mutually trusted KGC.

There are other distributed group key management protocols based on non-DH key agreement approach. Tzeng [33] proposed a conference key agreement protocol based on discrete logarithm (DL) assumption with fault tolerance in recent years. However, there is a serious encumbrance to efficiency. In 2008, Cheng and Laih [12] modified Tseng's conference key agreement protocol based on bilinear pairing. In 2009, Huang et al. [17] proposed a noninteractive protocol based on DL assumption to improve the efficiency of Tseng's protocol. One main concern of key agreement protocols is that since all communication users are involved to determine session keys, the time delay of setting up this group key may be too long, especially when there are a large number of group members.

Since avoiding the use of encryption one by one can introduce less computation complexity, secret sharing has been used to design group key distribution protocols, which was first introduced by both Blakley [1] and Shamir [29] independently in 1979. There are two different approaches using secret sharing: one assumes a trusted offline server active only at initialization [4], [15], [28], [3] and the other assumes an online trusted server, called the key generation center (KGC), always active. The first type of approach is also called the key predistribution scheme. The main disadvantage of this approach is to require every user to store a large size of secrets. The second type of approach requires an online server to be active [24]. It is similar to the model used in the IEEE 802.11i standard [20]. In 1989, Laih et al. [24] proposed the first algorithm based on this approach using any (t, n) secret sharing scheme to distribute a group key to a group consisting of $(t-1)$ members. Later, there are some papers [2], [25], [28] following the same concept to propose ways to distribute group messages to multiple users. Recently, [18] proposed a group key transfer protocol using (t, n) secret sharing that provided confidentiality and authentication, where KGC and each group member need to compute a t-degree interpolating polynomial to encrypt and decrypt the secret group key respectively. Then [26] pointed out that [18] could not protect users' long-term secrets against a malicious user and further gave an improvement. The main disadvantage of the approach of relying on an online KGC is that the trusted KGC is required in distributing the group key and it increases the overhead of system.

Due to the fact that linear secret sharing can be seen as a natural and useful generalization of threshold secret sharing and has an advantage in terms of computational complexity (that is, there is no need to compute a t-degree interpolating polynomial to encrypt and decrypt the secret), linear secret sharing schemes (LSSS) have received considerable attention and research in this field.

In this paper, we first adopt the advantages of DH key agreement and LSSS to design a secure and efficient key transfer protocol without an online KGC. The confidentiality of the group key transfer phase of this protocol is information theoretically secure, which is ensured by a perfect LSSS. We classify attacks into insider and outsider attacks separately, and analyze our protocol under these attacks in detail.

The rest of this paper is organized as follows: In the next section, we provide some preliminaries. In Section 3, we describe our main objective. In Section 4, we propose our group key transfer protocol. We analyze the security of our proposed protocol in Section 5. We conclude in Section 6.

2 Preliminaries

In this section we review some basic definitions concerning CDH assumption and linear secret sharing schemes.

2.1 CDH Assumption

Definition 1 (The Computational Diffie-Hellman (CDH) Assumption). Let $G = \langle g \rangle$ be a multiplicative cycle group of order q, and two integers a, b are chosen in $\mathbb{Z}_q^*$. Given g, g^a and g^b, a adversary has a negligible success probability ε for obtaining an element $\phi \in G$, such that $\phi = g^{ab}$ within polynomial time.

2.2 Linear Secret Sharing Schemes and Monotone Span Programs

Let $\mathcal{P} = \{1, ..., n\}$ be the set of participants. An access structure, denoted by Γ, is a collection of subsets of $\mathcal{P}$ satisfying the monotone ascending property: for any $A' \in \Gamma$ and

$A \in 2^{\mathcal{P}}$, $A' \subseteq A$ implies $A \in \Gamma$. An adversary structure, denoted by $\mathcal{A}$, is a collection of subsets of $\mathcal{P}$ satisfying the monotone descending property: for any $A' \in \mathcal{A}$ and $A \in 2^{\mathcal{P}}$, $A \subseteq A'$ implies $A \in \mathcal{A}$. Because of the monotone properties, for any access structure Γ and any adversary structure $\mathcal{A}$, it is enough to consider the minimum access structure $\Gamma_{\min} = \{A \in \Gamma \mid \forall B \subset A \Rightarrow B \notin \Gamma\}$ and the maximum adversary structure $\mathcal{A}_{\max} = \{B \in \mathcal{A} \mid \forall A \supset B \Rightarrow A \notin \mathcal{A}\}$ respectively. Generally, we deal with the complete situation, i.e., $\mathcal{A} = 2^{\mathcal{P}} - \Gamma$. In this paper, we consider a specific access structure used for group key distribution, that is, $\Gamma_{\min} = \Gamma = \{\mathcal{P}\}$.

Suppose that S is the secret-domain and P_i is the share-domain of participant i, where $1 \le i \le n$. When a dealer D wants to share a secret $s \in S$ among a set of participants $\mathcal{P} = \{1, ..., n\}$, he will give each participant a share $p_i \in P_i$. The shares should be distributed secretly, so no participant knows the share given to another participant. At a later time, a subset of participants will attempt to reconstruct the secret s from the shares they collectively hold. By using Shannon's entropy function, a secret sharing scheme with respect to an access structure Γ is defined such that the following requirements are satisfied[19].

i) Correctness requirement: any subset $A \subseteq \mathcal{P}$ of participants enabled to recover s can compute s. Formally, for all $A \in \Gamma$, it holds $H(S \mid A) = 0$.

ii) Security requirement: any subset $A \subseteq \mathcal{P}$ of participants not enabled to recover s, even pooling all of their shares together, can not reconstruct s. Formally, for all $A \notin \Gamma$, it holds $0 < H(S \mid A) \le H(S)$.

In the security requirement, if for any $A \notin \Gamma$ it holds $H(S \mid A) = H(S)$ (that is, participants in A pool their shares together obtain no information on s), we call it a *perfect* secret sharing scheme which we are interested in. If $|S| = |P_i|$ for $1 \le i \le n$, then the secret sharing scheme is called *ideal*. Furthermore, a perfect secret sharing scheme is *linear*, if $S = \mathcal{K}$ is a finite field, P_i are linear spaces over $\mathcal{K}$ and the reconstruction operations are linear [10].

Karchmer and Wigderson [23] introduced monotone span programs (MSP) as linear models computing monotone Boolean functions. Usually we denote an MSP by $\mathcal{M}(\mathcal{K}, M, \psi)$, where M is a $d \times l$ matrix over a finite field $\mathcal{K}$ and $\psi : \{1,...,d\} \to \{1,...,n\}$ is a surjective labeling map which actually distributes to each participant some rows of M. We call d the size of the MSP. For any subset $A \subseteq \mathcal{P}$, there is a corresponding characteristic vector $\vec{\delta}_A = (\delta_1,...,\delta_n) \in \{0,1\}^n$ where for $1 \le i \le n$, $\delta_i = 1$ if and only if $i \in A$. Consider a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ which satisfies that for any $A \subseteq \mathcal{P}$ and $B \subseteq A$, $f(\vec{\delta}_B) = 1$ implies $f(\vec{\delta}_A) = 1$. We say that an MSP $\mathcal{M}(\mathcal{K}, M, \psi)$ computes the monotone Boolean function f with respect to a target vector $\vec{v} \in \mathcal{K}^l \setminus \{(0,...0)\}$, if it holds that $\vec{v} \in span\{M_A\}$ if and only if $f(\vec{\delta}_A) = 1$, where M_A consists of the rows r of M with $\psi(r) \in A$ and $\vec{v} \in span\{M_A\}$ means that there exists a vector $\vec{w}$ such that $\vec{v} = \vec{w} M_A$.

Beimel [10] proved that devising a linear secret sharing scheme (LSSS) for an access structure Γ is equivalent to constructing an MSP computing the monotone Boolean function f_Γ which satisfies $f_\Gamma(\vec{\delta}_A) = 1$ if and only if $A \in \Gamma$. On the other hand, an MSP $\mathcal{M}(\mathcal{K}, M, \psi)$ can compute f_Γ if and only if there exists a vector $\vec{v}$ which lies in the space $\bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i - \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$, where V_i is the space spanned by the row vectors of M distributed to participant i according to ψ and the vector $\vec{v}$ can be seemed as the target vector described above. Hence, finding the linear spaces V_i with the condition $\bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i - \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i \neq \varnothing$ is the key point of building an LSSS with respect to Γ.

3 Design Principles

In this section, we describe the model of our group key transfer protocol and the security goals for our group transfer protocol.

3.1 Model

The conventional group key transfer protocols based on threshold secret sharing schemes (TSSS) rely on an online, trusted KGC as the dealer to issue the shares (shadows) for each members and generate a secret key as the group key, which is then transported to each member by TSSS. Actually, this approach can result in loss of flexibility and cause an increase of the overhead associated with the implementation of system. To overcome these drawbacks, an initiator, one of the group members, is endowed with the authority to select a secret key as the group key and to originate the group communications. In addition, the initiator must share secrets (shadows) with the other members by using a secure and efficient method.

It is well known that the interactive key agreement protocol can construct a one-time secret between two parties in public environments. In our design, the concept of DH key agreement protocol is used to share secrets between the initiator and the other members of the group. These secrets determine a group of linearly independent vectors, where the number of these vectors is equal to the number of group members minus one. Further, the initiator can select a session key and separately compute the inner products of these vectors and a random vector determined by all group members. Afterwards, the initiator publishes each value of the session key minus each inner product, where the number of those public values is equal to the number of group members minus one. On the other hand, each group member except the initiator is able to use his/her secret and the related public value to reconstruct the session key. Finally, all group members share a common session key for group communications.

3.2 Security Goals

The main security goals for our group key transfer protocol are: 1) key freshness; 2) key confidentiality; and 3) key authentication.

Key freshness is to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication. Key confidentiality is to protect the group key such that it can only be recovered by authorized group members; but not by any un-authorized user. Key authentication is to provide assurance

to authorized group members that the group key is distributed by the initiator; but not by an attacker.

4 The Proposed Protocol

Our group key transfer protocol consists of two phases: the secret establishment phase and the session key transfer phase. Suppose that a set of n participants, $\{1,...,n\}$, wants to set up a secure communication. Each participant must maintain a public/private key pair (puk, prk), such that $puk = g^{prk} \bmod p$, where $g \in \mathbb{Z}_p^*$, p is a large, safe prime number. Note that the long-term pair (puk, prk) is authenticated by a trusted authority with the corresponding certificate. Suppose that an initiator, one of the group members, is n and endowed with the authority to select a secret key as the group key and to originate the group communications. The secret establishment phase contains the following steps:

- Step 1. The initiator broadcasts a request containing a random number $r_n \in \mathbb{Z}_p^*$, his/her long-term public key puk_n, and a list of members, $\{1,...,n\}$, to announce the group communication.
- Step 2. Upon receiving the announcement from the initiator, each group member i, for $i = 1,...,n-1$, selects a random number $r_i \in \mathbb{Z}_p^*$ and uses his/her private key prk_i to compute the secret as $s_i = puk_n^{\,prk_i r_i r_n} \bmod p$. Afterwards, i computes $Auth_i = h(s_i \,||\, r_n)$ and sends $\{r_i, puk_i, Auth_i\}$ to the initiator as a response.
- Step 3. After receiving the message from each i, the initiator computes $s_i^* = puk_i^{\,prk_n r_i r_n} \bmod p$ and then checks $Auth_i \stackrel{?}{=} h(s_i^* \,||\, r_n)$. If the result is valid, the initiator believes that the secret $s_i = g^{puk_i\, prk_n r_i r_n} \bmod p$ is shared with corresponding group member i. Otherwise, the initiator claims that i is fraudulent and then restarts the protocol.

In the session key transfer phase, we suppose that $\overline{V} = \mathcal{K}^n$ is the n dimensional linear space over $\mathcal{K}$, where $\mathcal{K}$ is a finite field with the characteristic $char(\mathcal{K}) = p$. Given a

basis $\{e_1,...,e_n\}$ of $\bar{V}$ with $\vec{e}_j = (0,...,0,\overset{j}{1},0,...,0) \in \mathcal{K}^n$ for $j = 1,...,n$, the mapping $\mathbf{v}: \mathcal{K} \to \bar{V}$ defined by $\mathbf{v}(x) = \sum_{j=1}^{n} x^{j-1} e_j$ is determined. Upon sharing the secret with corresponding group member $i (i \in \{1,...,n-1\})$, the initiator needs to randomly select a group key. The initiator needs to distribute this group key to the other group members in a secure and authenticated manner. All communications between the initiator and the other group members are in a broadcast channel. The initiator n and the other group members $i (i \in \{1,...,n-1\})$ execute the following steps:

- Step 1. The initiator separates each shared secret s_i into two parts x_i and y_i, where $(x_i \| y_i) = s_i$ for $i = 1,...,n-1$, and randomly generates a session key $K_G \in \mathbb{Z}_p^*$. Then, the initiator computes $n-1$ additional values, $U_i = (K_G - K_i) \bmod p$, for $i = 1,...,n-1$, and the value $Auth = h(K_G, 1,...,n, r_1,...,r_n, U_1,...,U_{n-1})$, where the vector $\vec{r} = (r_i,...,r_n) \in \mathcal{K}^n$, the inner product $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$ and h is a one-way hash function. The initiator broadcasts $\{Auth, U_i\}$, for $i = 1,...,n-1$, to the other group members. All computations are performed in $\mathcal{K}$.
- Step 2. For each group member except the initiator, $i (i \in \{1,...,n-1\})$, knowing the public value, U_i, is able to compute the inner product $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$ and recover the group key $K_G = (U_i + K_i) \bmod p$. Then, $i (i \in \{1,...,n-1\})$ needs to compute $h(K_G, 1,...,n, r_1,...,r_n, U_1,...,U_{n-1})$ and check whether this hash value is identical to $Auth$. If these two values are identical, i authenticates the group key is sent from the initiator.

After the above steps have been executed successfully, the session key K_G is established among all group members. Later, the key K_G can be used for secure group communications.

Remark 1. In our protocol, based on the CDH assumption, the initiator, n, shares a secret, s_i, with each user $i (i \in \{1,...,n-1\})$. Adding/removing any user does not need to update any existing shared secret. However, for distributing a secret group key involving n

group members, the initiator needs to broadcast a message containing n elements to all the other group members. At the same time, each group member except the initiator needs to compute the inner product $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$ and recover the group key $K_G = (U_i + K_i) \bmod p$, where the vector $\vec{r} = (r_i, ..., r_n) \in \mathcal{K}^n$. It is easy to see that the equation used to recover the group key is a function of each group member's random number and the secret shared between corresponding group member and the initiator. Observe that the vectors $\mathbf{v}(x)$ have Vandermonde coordinates with respect to the given basis of $\overline{V}$. This implies that every set of at most n vectors of the form $\mathbf{v}(x)$ is independent, that is, any less than or equal to n vectors of the form $\mathbf{v}(x)$ are linearly independent, which can not be represented by each other. Thus, the LSSS in our protocol is a perfect LSSS, which will be described later.

5 Security Analysis

In this section, we first prove the proposed LSSS in our protocol is a perfect LSSS. Then, we consider two types of adversaries in our proposed protocol, insider and outsider. Finally, we prove that our proposed protocol achieves the security goals mentioned in Section 3 and is against inside and outside attacks.

5.1 A Perfect and Ideal LSSS

We now prove that the proposed LSSS in our protocol is a perfect LSSS, in which a dealer D shares a secret, the inner product $(\vec{v}, \vec{r}) = s$ with $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i)$, among a set of participants $\mathcal{P} = \{1, ..., n\}$ and gives each participant i a share, the inner product $(y_i \mathbf{v}(x_i), \vec{r})$, such that for $\Gamma = \{\{1, 2, ..., n\}\}$ any $A \in \Gamma$ pool their shares together can compute s and any $A \notin \Gamma$ pool their shares together obtains no information on s. Firstly, we prove the following Proposition.

Proposition 1. Suppose that a group $\{1, ..., n\}$ consists of n members, $\Gamma = \{\{1, 2, ..., n\}\}$ and $\mathcal{K}$, x_i, y_i, $\overline{V}$, $1 \le i \le n$ are given as above. Let $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i)$ and

$V_i = span\{\mathbf{v}(x_i)\}$ for $1 \le i \le n$, then it holds that $\vec{v} \in \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i - \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$.

Proof. Firstly, we prove that $\vec{v} \in \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i$. Observe that $V_i = span\{\mathbf{v}(x_i)\}$ for $1 \le i \le n$ and $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i)$. These imply that there must exist a linear combination of the vectors in $\sum_{1 \le i \le n} V_i$ such that it equals to $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i)$. Namely, $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i) \in \sum_{1 \le i \le n} V_i$. Since $\Gamma_{\min} = \Gamma = \{\{1, 2, ..., n\}\}$, it implies that $\sum_{1 \le i \le n} V_i = \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i$. Hence, we obtain that $\vec{v} \in \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i$.

Then we prove that $\vec{v} \notin \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$. Due to the fact that $\mathbf{v}(x_i)$ $(1 \le i \le n)$ is the form $\mathbf{v}(x)$ and every set of at most $(n+1)$ vectors of the form $\mathbf{v}(x)$ is independent, we obtain that $\mathbf{v}(x_1), ..., \mathbf{v}(x_n)$ are linearly independent. Furthermore, observe that $V_i = span\{\mathbf{v}(x_i)\}$ for $1 \le i \le n$ and $\{1, 2, ..., n\} \notin \mathcal{A}_{\max}$. These imply that for every $B \in \mathcal{A}_{\max}$, there does not exist a linear combination of the vectors in $\sum_{i \in B} V_i$ such that it equals to $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i)$. Namely, $\vec{v} = \sum_{1 \le i \le n} y_i \mathbf{v}(x_i) \notin \sum_{i \in B} V_i$ for every $B \in \mathcal{A}_{\max}$. Hence, $\vec{v} \notin \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$.

As a consequence of the above, it holds that

$$\vec{v} \in \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i - \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i .$$

Theorem 1. The proposed LSSS in our protocol is a perfect LSSS.

Proof. From Proposition 1, seeing that $\vec{v} \in \bigcap_{A \in \Gamma_{\min}} \sum_{i \in A} V_i$, it implies that all users in the group $\{1, 2, ..., n\}$ pool their shares together can reconstruct the secret $(\vec{v}, \vec{r}) = s$ by computing a linear combination of their shares. Hence, it holds that $H(s \mid A) = 0$ for $A \in \Gamma = \{\{1, 2, ..., n\}\}$.

At the same time, from Proposition 1, seeing that $\vec{v} \notin \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$, it implies that any subset $B \notin \Gamma$ of users pool their shares together obtain no information on the secret $(\vec{v}, \vec{r}) = s$. Due to the fact that $\mathbf{v}(x_1), ..., \mathbf{v}(x_n)$ are linearly independent, furthermore, $V_i = span\{\mathbf{v}(x_i)\}$ for $1 \le i \le n$ and $\{1, 2, ..., n\} \notin \mathcal{A}_{\max}$, we obtain that $\vec{v} \notin \bigcup_{B \in \mathcal{A}_{\max}} \sum_{i \in B} V_i$.

It implies that there does not exist a linear combination of their shares such that it equals to s. Hence, for any $B \notin \Gamma = \{\{1, 2, ..., n\}\}$ it holds that $H(s \mid B) = H(s)$.

Therefore, according to definition in Section 2, the proposed LSSS is a perfect LSSS. Since there has no other computational assumption based upon, this LSSS is information theoretically secure.

Furthermore, it is easy to see that the proposed LSSS is an ideal LSSS. This means that its efficiency is optimal.

5.2 Attacks

Adversaries can be categorized into two types. The first type of adversaries is outsiders of a particular group. The outside attacker can try to recover the secret group key belonging to a group that the outsider is unauthorized to know. This attack is related to the confidentiality of group key. In our proposed protocol, the initiator broadcasts a request to announce the group communication. The outside attacker may also impersonate the initiator to request a group key service. In security analysis, we will show that the outside attacker gains nothing from this attack since the attacker cannot obtain the one-time shared secret. The second type of adversaries is insiders of a group who are authorized to know the secret group key; but inside attacker attempts to retrieve the previous secret between the other members and the initiator. Since any insider of a group is able to recover the same group key, we need to prevent inside attacker knowing the one-time shared secret used in each session.

The following theorem proves that our protocol can achieve the security goals we set previously.

Theorem 2. The proposed protocol achieves the following security goals: 1) key freshness, 2) key confidentiality, and 3) key authentication.

Proof. We assume that a group consists of n members, $\{1, ..., n\}$, and the one-time shared secrets are $s_1, ..., s_{n-1}$. The proposed protocol achieves the following security goals:

1) Key freshness is ensured by the initiator since a random group key is selected by the initiator for each request. In addition, the equation $K_G = (U_i + K_i) \bmod p$ used to recover the group key is a function of random number selected by each group member and the

one-time secret shared between corresponding group member and the initiator.

2) Key confidentiality is provided due to the security features of CDH assumption and the proposed perfect LSSS. By employing the proposed LSSS, the initiator randomly selects a group key K_G and makes $n-1$ values, $U_i = (K_G - K_i) \bmod m$ for $i = 1, ..., n-1$, publicly known. For each authorized group member, including the one-time secret s_i shared with the initiator, he/she knows the inner product $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$. Thus, any authorized group member is able to recover the secret group key $K_G = (U_i + K_i) \bmod m$. However, for any unauthorized member (or outsider), there are only $n-1$ values $U_i = (K_G - K_i) \bmod m$ for $i = 1, ..., n-1$ available and he obtains no information on K_i and $\sum_{1 \le i \le n-1} K_i$. Thus, unauthorized member knows nothing about the group key. In other words, the proposed perfect LSSS is information theoretically secure, so the group key transfer phase of our protocol is also information theoretically secure.

At the same time, the one-time secret s_i generated by an interactive key agreement protocol with random numbers, and then the shared secret s_i is used to construct the current group key. Even though the current group key is compromised, it does not reveal any information regarding the previous group keys. Therefore, our protocol achieves forward secrecy.

3) Key authentication is provided through the values $Auth_i$ and $Auth$. In step 2 of the secret establishment phase, $Auth_i$ is a one-way hash output with the shared secret s_i and the initiator's random number as input. Since the shared secret s_i is known only to corresponding group member i and the initiator, unauthorized members cannot forge this value. In addition, any replay of $\{r_i, puk_i, Auth_i\}$ of the group member i in step 2 of the secret establishment phase can be detected since the shared secret s_i is a function of the group member i's and the initiator's random numbers.

In step 1 of the group key transfer phase, $Auth$ is a one-way hash output with the secret

group key and all members' random numbers as input. Since the group key is known only to authorized group members and the initiator, unauthorized members cannot forge this value. Any insider also cannot forge a group key without being detected since the group key is a function of the secret shared between each group member and the initiator. In addition, any replay of $\{Auth, U_i\}$, for $i = 1, ..., n-1$, of the initiator in step 1 of the group key transfer phase can be detected since the group key is a function of each group member's random number.

Theorem 3 (Outsider attack). Assume that an attacker wants to masquerade as a group member to join the group communication, then, the attacker can neither obtain the group key nor share a group key with any group member.

Proof. Although any attacker can intercept the messages between the initiator and any other group member i, the attacker cannot share the one-time secret $s_i = puk_n^{prk_i r_i r_n} \bmod p$ with the initiator successfully, due to the fact that the long-term private key prk_i of any member i is unknown. In addition, the group key K_G, which is constructed by using the proposed LSSS, can only recovered by any honest member who has the correct corresponding shared secret s_i. Therefore, the attacker cannot masquerade as any group member to obtain the group key K_G by intercepting messages. On the other hand, since the attacker does not have the private key prk_n of the initiator, thus the attacker cannot masquerade as the initiator successfully to share the secret s_i with the other members. In other words, the attacker cannot share the key K_G with any group member by masquerading as the initiator.

If the attacker tries to reuse a compromised group key by replaying previously recorded key information from the initiator, this attack cannot succeed in sharing this compromised group key with any group member since the group key is a function of each member's random number and the one-time secret shared between the initiator and each other group member. A compromised group key cannot be reused if each member selects a random number for every group communication.

Theorem 4 (Insider attack). Assume that the protocol runs successfully many times,

then the one-time secret $(x_i \| y_i) = s_i$ of each i shared with the initiator remains unknown to all other group members (and outsiders).

Proof. In order to transfer the group key, the initiator randomly selects a group key K_G and makes $n-1$ values, $U_i = (K_G - K_i) \bmod p$ for $i = 1, ..., n-1$, publicly known. For each authorized group member, with knowledge of the one-time secret shared with the initiator and the public information, he/she knows U_i and is able to compute the inner product $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$. Thus, any authorized group member is able to reconstruct the group key $K_G = (U_i + K_i) \bmod p$, where the vector $\vec{r} = (r_i, ..., r_n) \in \mathcal{K}^n$. However, the one-time secret $(x_i \| y_i) = s_i$ of each group member shared with the initiator cannot be traced by outsiders.

For an insider j, he/she knows the group key and U_i, then he can obtain K_i from $K_G = (U_i + K_i) \bmod p$. However, j cannot solve the secret $(x_i \| y_i) = s_i$ from the equation $(y_i \mathbf{v}(x_i), \vec{r}) = K_i$ since there are two unknown quantities. At the same time, due to the fact that the secret $(x_i \| y_i) = s_i$ of each group member shared with the initiator depends on the random numbers (r_i, r_n) and long-term private keys (prk_i, prk_n), the one-time secret s_i is still untraceable by insiders.

Remark 2. Most key transfer schemes based on TSSS are claimed information theoretically secure. However, these schemes must pre-share secrets (shadows) between the dealer and the participants. It means that the secrets must be shared via a secure channel. Actually, it is a strong assumption to suppose that a secure channel is existed in public networks. That is, most existing schemes do not propose any practical method to share secrets in public networks. In this paper, we first used the CDH assumption to share the secrets between the initiator and other participants. Next, we construct a group key transfer protocol based on a perfect LSSS, which is no need to compute a t-degree interpolating polynomial to encrypt and decrypt the group key. This LSSS is information theoretically secure since there has no other computational assumption based upon. Hence, we say that the group key

transfer phase of our scheme is also information theoretically secure.

6 Conclusions

We have proposed an efficient group key transfer protocol without an online KGC, which is based on a perfect LSSS. The confidentiality of the group key transfer phase of this protocol is information theoretically secure. We provide group key authentication. Security analysis for possible attacks is included. As a result, this protocol can resist potential attacks and also reduce the overhead of system implementation. It is fairly interesting for practical applications.

References

[1] G.R. Blakley, “Safeguarding Cryptographic Keys,” Proc. Am. Federation of Information Processing Soc. (AFIPS ’79) Nat’l Computer Conf., vol. 48, pp. 313-317, 1979.

[2] S. Berkovits, “How to Broadcast a Secret,” Proc. Eurocrypt ’91 Workshop Advances in Cryptology, pp. 536-541, 1991.

[3] R. Blom, “An Optimal Class of Symmetric Key Generation Systems,” Proc. Eurocrypt ’84 Workshop Advances in Cryptology, pp. 335-338, 1984.

[4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly Secure Key Distribution for Dynamic Conferences,” Information and Computation, vol. 146, no. 1, pp. 1-23, Oct. 1998.

[5] C. Boyd, “On Key Agreement and Conference Key Agreement,” Proc. Second Australasian Conf. Information Security and Privacy (ACISP ’97), pp. 294-302, 1997.

[6] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, “Provably Authenticated Group Diffie-Hellman Key Exchange,” Proc. ACM Conf. Computer and Comm. Security (CCS ’01), pp. 255-264, 2001.

[7] E. Bresson, O. Chevassut, and D. Pointcheval, “Provably-Secure Authenticated Group Diffie-Hellman Key Exchange,” ACM Trans. Information and System Security, vol. 10, no. 3, pp. 255-264, Aug. 2007.

[8] J.M. Bohli, “A Framework for Robust Group Key Agreement,” Proc. Int’l Conf. Computational

Science and Applications (ICCSA ’06), pp. 355-364, 2006.

[9] M. Burmester and Y.G. Desmedt, “A Secure and Efficient Conference Key Distribution System,” Proc. Eurocrypt ’94 Workshop Advances in Cryptology, pp. 275-286, 1994.

[10] A. Beimel, “Secure schemes for secret sharing and key distribution,” Ph.D. dissertation, Technion—Israel Inst. Technol., Haifa, Israel, 1996.

[11] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, “Multicast Security: A Taxonomy and Some Efficient Constructions,” Proc. IEEE INFOCOM ’99, vol. 2, pp. 708-716, 1999.

[12] J.C. Cheng and C.S. Laih, “Conference Key Agreement Protocol with Non Interactive Fault-Tolerance Over Broadcast Network,” Int’l J. Information Security, vol. 8, no. 1, pp. 37-48, 2009.

[13] W. Diffie and M.E. Hellman, “New Directions in Cryptography,” IEEE Trans. Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.

[14] M. Eltoweissy, M.H. Heydari, L. Morales, and I.H. Sudborough, “Combinatorial Optimization of Group Key Management,” J. Network and Systems Management, vol. 12, no. 1, pp. 33-50, 2004.

[15] A. Fiat and M. Naor, “Broadcast Encryption,” Proc. 13th Ann. Int’l Cryptology Conf. Advances in Cryptology (Crypto ’93), pp. 480-491, 1994.

[16] H. Harney, C. Muckenhirn, and T. Rivers, “Group Key Management Protocol (GKMP) Architecture,” RFC 2094, July 1997.

[17] K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai, and T.S. Chen, “A Conference Key Agreement Protocol with Fault-Tolerant Capability,” Computer Standards and Interfaces, vol. 31, pp. 401-405, Jan. 2009.

[18] L. Harn, C. Lin, Authenticated Group Key Transfer Protocol Based on Secret Sharing, IEEE Trans. Computers, vol. 59, no. 6, pp. 842-846, June. 2010.

[19] Chingfang Hsu, Qi Cheng, Xueming Tang, Bing Zeng. An Ideal Multi-secret Sharing Scheme based on MSP. Information Sciences, Vol.181, No.7, 2011: 1403-1409

[20] IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.

[21] I. Ingemarsson, D.T. Tang, and C.K. Wong, “A Conference Key Distribution System,” IEEE Trans. Information Theory, vol. IT-28, no. 5, pp. 714-720, Sept. 1982.

[22] J. Katz and M. Yung, “Scalable Protocols for Authenticated Group Key Exchange,” J. Cryptology, vol. 20, pp. 85-113, 2007.

[23] M. Karchmer and A. Wigderson, “On span programs,” in Proc. 8th Annu. Conf. Structure in Complexity, San Diego, CA, May 1993, pp. 102–111.

[24] C. Laih, J. Lee, and L. Harn, “A New Threshold Scheme and Its Application in Designing the Conference Key Distribution Cryptosystem,” Information Processing Letters, vol. 32, pp. 95-99, 1989.

[25] C.H. Li and J. Pieprzyk, “Conference Key Agreement from Secret Sharing,” Proc. Fourth Australasian Conf. Information Security and Privacy (ACISP ’99), pp. 64-76, 1999.

[26] J Nam, M Kim, J Paik, W Jeon, B Lee and D Won, Cryptanalysis of a group key transfer protocol based on secret sharing, Future Generation Information Technology Third International Conference (FGIT 2011), Lecture Notes In Computer Science, Jeju Island, Korea, 2011: vol. 7105, 309-315

[27] A. Perrig, D. Song, and J.D. Tygar, “Elk, A New Protocol for Efficient Large- Group Key Distribution,” Proc. IEEE Symp. Security and Privacy, pp. 247-262, 2001.

[28] G. Saze, “Generation of Key Predistribution Schemes Using Secret Sharing Schemes,” Discrete Applied Math., vol. 128, pp. 239-249, 2003.

[29] A. Shamir, “How to Share a Secret,” Comm. ACM, vol. 22, no. 11, pp. 612-613, 1979.

[30] A.T. Sherman and D.A. McGrew, “Key Establishment in Large Dynamic Groups Using One-Way Function Trees,” IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.

[31] D.G. Steer, L. Strawczynski, W. Diffie, and M.J. Wiener, “A Secure Audio Teleconference System,” Proc. Eighth Ann. Int’l Cryptology Conf. Advances in Cryptology (Crypto ’88), pp. 520-528, 1988.

[32] M. Steiner, G. Tsudik, and M. Waidner, “Diffie-Hellman Key Distribution Extended to Group Communication,” Proc. Third ACM Conf. Computer and Comm. Security (CCS ’96), pp. 31-37, 1996.

[33] W.G. Tzeng, “A Secure Fault-Tolerant Conference Key Agreement Protocol,” IEEE Trans. Computers, vol. 51, no. 4, pp. 373-379, Apr. 2002.