A n Efficient Key Exchange Protocol Based on Secret Sharing

Guidename [R.Sailaja,]   Student name [P.Sujatha]

[1] Assoc Professor,Dept of Computer Science and Engineering Sri Sai Aditya Institute of Science and Technology Surampalem

[2] Final M.Tech Student, Dept of Computer Science and Engineering , Sri Sai Aditya Institute of Science and Technology Surampalem

_____

Abstract

Group key protocols are still an important research issue in the network security, generation and sharing the secret session key is the important factor during the the groupkey transfer protocols. Usually it will be taken care of KGC. KGC encrypts session keys under another secret key shared with each entity during registration. In this paper, we propose an authenticated key transfer protocol based on secret sharing scheme that KGC can broadcast group key information to all group members at once and only authorized group members can recover the group key but unauthorized users can not recover the groupkey.In this proposed mechanism shows More security levels In case of confidentiality and authentication than the traditional group key protocols can be analyzed in detail.

Index Terms- Group key transfer protocol, session key, secret sharing,confidentiality, authentication
_____

1. INTRODUCTION

Multicast key management, which is much different from unicast key management , is one of the most attractive area of cryptography. For an unicast application, the Diffie-Hellman key exchange protocol can be employed to establish a KEK (Key Encryption Key) between two entities. Then use this KEK to dispatch or update a session key. In contrast, the situation is much more complicated for a multicast application. A multicast application must dynamically handle multi-entities. For example, in a dynamic multicast group, the membership is changeable all the time due to frequently users' addition and eviction. Therefore, the key materials will probably be revealed if no security policies are adopted. For instance, if the key is not updated after the membership change, a new comer is able to read the contents before his coming, or a evictor is capable of reading the content after his leaving. In this case, multicast key management scheme should provide forward secrecy and backward secrecy for security reasons in some special applications, e.g. Pay-Per-View. In the past two decades, researchers have proposed many multicast key management schemes [5], [6], [7]. These schemes can be categorized into three different types: centralized, decentralized and distributed. A centralized group key management scheme involves a Key Server (KS) to generate and distribute shared key to all group members via a secure channel. A decentralized key management divides the whole group into smaller subgroups. Each subgroup is controlled by a single or several KS. A Distributed scheme allows each member to take part in a group key generation collaboratively. Each of the three schemes has its own advantages and disadvantages. Centralized scheme is the simplest one but has the risk of single-point-failure. Decentralized scheme adds some communication complexity between two members within different subgroups. Distributed scheme is somehow more complex than

the other two, but it doesn't involve KS. This feature is very useful in the case of no one can play the role of KS, e.g. a sensor Ad-hoc network application.

Secret sharing has been used to design group key distribution protocols. There are two different approaches using secret sharing: one assumes a trusted offline server active only at initialization [4], [14], [25], [3] and the other assumes an online trusted server, called the key generation center, always active. The first type of approach is also called the key predistribution scheme. In a key predistribution scheme, a trusted authority generates and distributes secret pieces of information to all users offline. At the beginning of a conference,users belonging to a privileged subset can compute individually a secret key common to this subset. A family of forbidden subsets of users must have no information about the value of the secret. 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 [20] and this approach is similar to the model used in the IEEE 802.11i standard [17] that employs an online server to select a group key and transport it to each group member. However, the difference between this approach and the IEEE 802.11i is that, instead of encrypting the group temporal key (GTK) using the key encryption key (KEK) from the authentication server to each mobile client separately as specified in the IEEE 8-2.11i, the trusted KGC broadcasts group key information to all group members at once. In 1989, Laih et al. [20] proposed the first algorithm based on this approach using any $ðt; nÞ$ secret sharing scheme to distribute a group key to a group consisting of $ðt \_ 1Þ$ members. Later, there are some papers [2], [21], [25] following the same concept to propose ways to distribute group messages to multiple users. In this paper, we propose a solution based on this approach and provide confidentiality and authentication for distributing group keys. Furthermore, we classify attacks into insider and outsider attacks separately, and analyze our protocol under these attacks in detail.

We list following unique features of our proposed group key transfer protocol using secret sharing scheme.

- Each user needs to register at KGC to subscribe the group key transfer service and to establish a secret with KGC. Thus, a secure channel is needed initially to share this secret with each user. Later, KGC can transport the group key and interact with all group members in a broadcast channel.
- The confidentiality of group key distribution is information theoretically secure; that is, the security of this transfer of group key to each group member does not depend on any computational assumption.

  The authentication of the group key is achieved by broadcasting a single authentication message to all group members.

2. LITERATURE SURVEY

Definition 1 (Factoring Problem). Let us choose two large safe primes p and q (i.e., primes such that $p0 ¼ p\_1 2$ and $q0 ¼ q\_1 2$ are also primes) and compute $n ¼ pq$. n is made publicly known. Factoring problem is defined to compute factors p and q such that $n ¼ pq$.

Definition 2 (Factoring Assumption). It is computationally intractable to solve the Factoring Problem.

Secret sharing schemes were introduced by both Blakley [1] and Shamir [26] independently in 1979 as a solution for safeguarding cryptographic keys and have been studied extensively in the literatures. In a secret sharing scheme, a secret s is divided into n shares and shared among n shareholders in such a way that, with any t or more than t shares, it is able to reconstruct this secret; but, with fewer than t shares, it cannot reconstruct the secret. Such a scheme is called a ðt; nÞ secret sharing, denoted as ðt; nÞ-SS. Shamir's ðt; nÞ-SS. In Shamir's ðt; nÞ-SS [26] based on Lagrange interpolating polynomial, there are n shareholders U ¼ fU1; . . . ; Ung and a mutually trusted dealer D. The scheme consists of two algorithms:

1. Share generation algorithm: dealer D does the following:.

- dealer D first picks a polynomial fðxÞ of degree ðt _ 1Þ randomly: fðxÞ ¼ a0 þ a1xþ _ _ _þat_1xt_1, in which

the secret s ¼ a0 ¼ fð0Þ and all coefficients a0; a1; . . . ; at_1 are in a finite field IFp ¼ GFðpÞ with p elements.

- D computes all shares: si ¼ fðiÞðmod pÞ for i ¼ 1; . . . ; n.
- Then, D outputs a list of n shares ðs1; s2; . . . ; snÞ and distributes each share si to corresponding shareholder Pi privately.

2. Secret reconstruction algorithm: this algorithm takes any t shares ðsi1 ; . . . ; sit Þ as input, it can reconstruct the
secret s as

$$s = f(0) = \sum_{i \in A} s_i \beta_i$$

$$= \sum_{i \in A} s_i \left(\prod_{j \in A-\{i\}} \frac{x_j}{x_j - x_i} \right) \pmod{p},$$

where $A = \{i_1, \ldots, i_t\} \subseteq \{1, 2, \ldots, n\}$, β_i for $i \in A$ are
Lagrange coefficients.

Lagrange coefficients. We note that the above scheme satisfies the basic security requirements of secret sharing scheme as follows: 1) with knowledge of any t or more than t shares, it can reconstruct the secret s easily; and 2) with knowledge of fewer than ðt _ 1Þ shares, it cannot reconstruct the secret s. Shamir's scheme is information theoretically secure since the scheme satisfies these two requirements without making any computational assumption. For more information on this scheme, readers can refer to the original paper [26]. In Shamir's ðt; nÞ-SS, the secret of each shareholder is just the y-coordinate of fðxÞ and the x-coordinate is made publicly known. However, in our proposed group key transfer protocol, for security reason, we need to keep both x-coordinate and y-coordinate as each user's secret. Furthermore, in Shamir's ðt; nÞ-SS, the modulus p used for all computations is a prime number. In our proposed protocol, to prevent insider attack as we will explain this later, the modulus n used for computations is a composite integer. We should point out that finding a modular inverse is needed in secret reconstruction process. We can use Euclid's extended algorithm [30] to compute modular inverse without factoring the composite modulus n.

3. PROPOSED PROTOCOL

Prototype :

  Group key transfer protocol relies on one trusted entity, KGC, to choose the key, which is then transported to each member involved. Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret with each user. In most key transfer protocol, KGC encrypts the randomly selected group key under the secret shared with each user during registration and sends the ciphertext to each group member separately. An authenticated message checksum is attached with the ciphertext to provide group key authenticity. In this approach, the confidentiality of group key is ensured using any encryption algorithm which is computationally secure. Our protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition, the authentication of broadcasting
message can be provided as a group authentication. This feature provides efficiency of our proposed protocol.

    Our authenticated group key transfer protocol consists of three processes: initialization of KGC, user registration, and group key generation and distribution. The detailed description is as follows:

Initialization of KGC. The KGC randomly chooses two safe primes p and q (i.e., primes such that $p^1=(p-1)/2$ and $q^1= (q-1)/2$ are also primes) and compute n = pq. n is made publicly known.

User Registration. Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret, (x_i,y_i), with each user Ui, where x_i, $y_i \in Z_n$
.
Group key generation and distribution. Upon receiving a group key generation request from any user, KGC needs to randomly selects a group key and access all shared secrets with
group members. KGC needs to distribute this group key to all group members in a secure and authenticated manner. All communication between KGC and group members are in a broadcast channel. For example, we assume that a group consists of t members, fU1; U2; . . . ; Utg, and shared secrets are (x_i,y_i), for i ¼ 1; . . . ; t. The key generation and distribution process contains
five steps.

Step 1. The initiator sends a key generation request to KGC with a list of group members as {U1; U2; . . . ; Ut}.

Step 2. KGC broadcasts the list of all participating members, {U1; U2; . . . ; Ut}, as a response.

Step 3. Each participating group member needs to send a random challenge, R_i E Z^+n, to KGC.

Step 4. KGC randomly selects a group key, k, and generates an interpolated polynomial fðxÞ with degree to pass through (t+1) points, (0,k) and (x_i,y_i XOR R_i), for i = 1; . . . ; t. KGC also computes t additional points, Pi, for i =1; . . . ; t, on f(x) and Auth ¼=h(k;U1; . . . ; Ut; R2; . . .;Rt; P1; . . . ; Pt), where h is a one-way hash function. All computations on f(x) are over Z^+_n. KGC broadcasts {Auth; Pi, for i = 1; . . . ; t, to all group members. All computations are performed in Z E n.

Step 5. For each group member, Ui, knowing the shared secret, (x_i,y_i XOR R_i),, and t additional public points, Pi, for i = 1; . . . ; t, on f(x), is able to compute the polynomial fðxÞ and recover the group key k =f{0}. Then, Ui computes h(k, U1; . . . ; Ut;R1; . . .; Rt; P1; . . . ; Pt) and checks whether this hash value is identical to Auth. If these two values are identical, Ui authenticates the group key is sent from KGC.

In Fig. 1, we illustrate this group key transfer protocol for a group containing three members, A, B, and C.. In our protocol, during registration, KGC shares a secret, ðxi; yiÞ, with each user Ui. Adding/removing any user does not need to update any existing shared secret. However, for distributing a secret group key involving t group members, KGC needs to broadcast a message containing ðt þ 1Þ elements to all group members. At the same time, each group member needs to compute a t-degree interpolating polynomial fðxÞ to decrypt the secret group key. Thus, our proposed protocol is only suitable for distributing secret group key to a group with a small group size. If a group containing a large group size, such as applications in pay-per-view system, centralized group key distribution protocols, such as EBS protocol [13], can be used to reduce the length of broadcast message and computational load of each group member
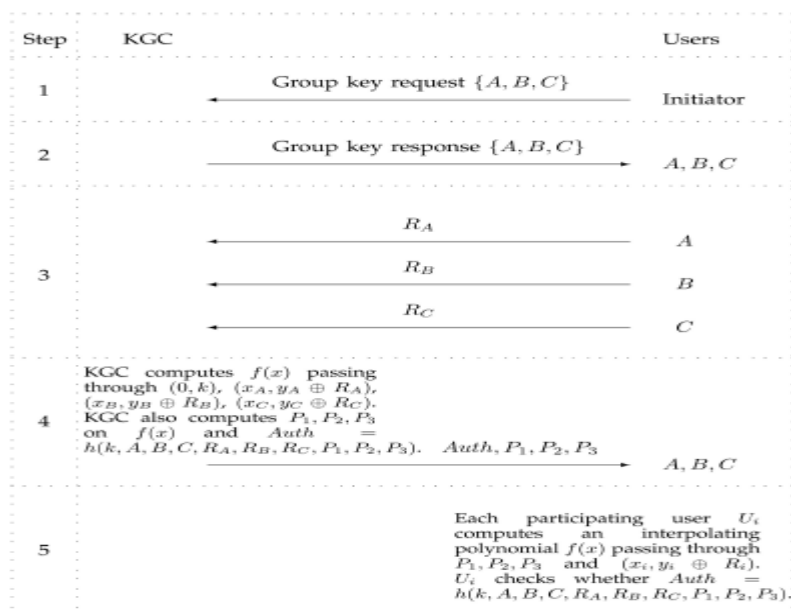
Group key transfer protocol

4.SECURITY ANALYSIS

Attacks :
4.1 Attacks
Adversaries can be categorized into two types. The first type of adversaries are 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, anyone can send a request to KGC for requesting a group key service. The outside attacker may also impersonate a group user 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 recover the group key. The second type of adversaries are insiders of a group who are authorized to know the secret group key; but inside attacker attempts to recover other member's secret shared with KGC. Since any insider of a group is able to recover the same group key, we need to prevent inside attacker knowing other member's secret shared with KGC.

Outsider attack :
        Assume that an attacker who impersonates a group member for requesting a group key service, then the attacker can neither obtain the group key nor share a group key with any group member Although any attacker can impersonate a group member to issue a service request to KGC without being detected and KGC will respond by sending group key information accordingly; however, the group key can only be recovered by any group member who shares a secret with KGC. This security feature is information theoretically secure. If the attacker tries to reuse a compromised group key by replaying previously recorded key information from KGC, 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 challenge and the secret shared between group member and KGC. A compromised group key cannot be reused if each member selects a random challenge for
every conference.

Insider attack :
Assume that the protocol runs successfully v times and the applied factoring instances are intractable, then the secret (x_i, y_i) of each group member shared with KGC remains unknown to all other group members (and outsiders).
For a group key service request, KGC generates a tth degree polynomial $f(x)$ passing through $(t + 1)$ points, $(x_i, y_i) \in R_i$, for i = 1, . . . , t. For each authorized group member, with knowledge of the secret shared with KGC and t public information, he/she knows $(t + 1)$ points on $f(x)$. Thus, any authorized group member is able to reconstruct the polynomial $f(x)$. However, the secret (x_i, y_i) of each group member shared with KGC remains unknown to outsiders.

5.CONCLUSION:

   In this paper we proposed a novel mechanism for group key transfer protocol based on secret sharing with the help of trusted KGC and preshare a secret with KGC and it broadcasts all the information slimantaniously.It provides the security measures like confidentiality and

authentication. We provide group key authentication. Security analysis for possible attacks is included.