AUTHENTICATED SEQUENCE OF SYSTEMS FOR RELIABLE BROADCAST USING KEY TRANSFER PROTOCOL

R.A.Karthika[1] , S. Maria Celestin Vigila[2]

[1]*M.E(Software Engineering), Noorul Islam University,*
[2]*Asst. Prof., Department of Information Technology, Noorul Islam University*
karthika78@gmail.com, [2]celesleon@yahoo.com

Abstract: **Group communication is becoming the basis for a growing number of applications. It is therefore critical to provide sound security mechanisms for group communication. Group communication can benefit from a Key Generation Center to achieve a scalable exchange of messages. Key Generation Center can broadcast group key information to all group members at once under the secret shared with each user during registration and sends the ciphertext to each group member separately. The strength of the proposed protocol is authentication and confidentiality. Security goals and analysis against inside and outside attacks can be achieved.**

Key Words: **Key transfer protocol, session key, secret sharing, sequence systems, group communication.**

I. INTRODUCTION

Providing authentication and confidentiality for the messages exchanged between group members is an important issue in Secure Group communication. To provide these two functions, one-time session keys need to be shared among communication entities to encrypt and authenticate messages. Thus, before exchanging communication messages, a key establishment protocol needs to distribute one-time secret session keys to all participating entities. Key establishment protocols are of two types: key agreement protocols and key transfer protocols.

In key agreement protocols, all communication entities are involved to determine session keys. The most commonly used key agreement protocol is Diffie-Hellman (DH) key agreement protocol. However, DH public key distribution algorithm can only provide session key for two entities; not for a group more than two members. One main concern of key agreement protocols is that since all communication entities 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.

When a secure communication involves more than two entities, a group key is needed for all group members. Group communication applications can use key transfer protocol to transmit data to all group members using minimum resources.

Efficiency is achieved because data packets need to be transmitted once and they traverse any link between two nodes only once, hence saving bandwidth. Key transfer protocols rely on a mutually trusted Key Generation Center (KGC) to select session keys and then transport session key to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration.

Secret sharing has been used to design group key distribution protocols. There are two different approaches using secret sharing: a trusted offline server called key predistribution scheme active only at initialization and an online trusted server, called the key generation center, which is always active

Key Predistribution Scheme:

In a key predistribution scheme, a trusted authority generates and distributes secret pieces common to this subset. The main disadvantage of this approach is to require every user to store a large size of secrets.

Online Server:

Online server needs to be active to select a group key and transport it to each group member. However, the difference between encrypting the Group Temporal Key (GTK) using the Key Encryption Key (KEK) from the authentication server to each mobile client separately is the trusted KGC broadcasts group key information to all group members at once.

The main security goals for our group key transfer protocol are: key freshness, key confidentiality and key authentication.

Freshness formalizes the fact that the session key is not obviously known by the adversary through basic means and to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication. On top of this and because the corruption capabilities of an adversary can make him learn the session key trivially, the definition is relevant to the notion of forward secrecy, which entails that the corruption of a player does not compromise the previously established session keys.

Key confidentiality is to protect the group key such that it can only be recovered by authorized group members; but not by any unauthorized user.

Key authentication is to provide assurance to authorized group members that the group key is distributed by KGC; but not by an attacker. An attacker can impersonate a user to request for a group key service. In addition, attacker can also modify information transmitted from users to KGC without being detected. Attackers can neither obtain the group key nor share a group key with authorized group members.

The proposed protocol is based on Exclusion Basis Systems (EBS), a combinatorial formulation of the group key management problem, which allows protocol user to trade-off between the number of keys needed to be stored and the number of messages needed to be transmitted for each key update with no counter collusion solution provided.

II. RELATED WORKS

Secret group communication can be achieved by encrypting messages with a group key. In a key predistribution scheme, 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 [1] to be active to select a group key, transport it to each group member and to encrypt GTK using the KEK from the authentication server to each mobile client separately.

We generalize the problem in which the secret is some data. Our goal is to divide D into n pieces D_1, ..., D_n in such a way that: knowledge of any k or more D_i pieces makes D easily computable; knowledge of any k-1 or fewer D_i pieces leaves D completely. Such a scheme is called a (k,n) threshold scheme. Threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate

To communicate securely over insecure channels it is essential that secret keys are distributed securely. Secret sharing scheme provides a solution for safeguarding cryptographic keys. 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 (t, n)-Secret Sharing, denoted as (t, n)-SS.

Shamir's (t, n)-SS is based on Lagrange interpolating polynomial. There are n shareholders $U=\{U_1 \ldots U_n\}$ and a mutually trusted dealer D. This scheme consists of two algorithms: share generation algorithm and secret reconstruction algorithm.

Share Generation Algorithm:

- Dealer D first picks a polynomial f(x) of degree (t-1) randomly:
  $f(x)=a_0+a_1x+\ldots.+a_{t-1}x^{t-1}$, in which the secret s is a constant term. $s=a_0=f(0)$ and all coefficients $a_0,a_1, \ldots ,a_{t-1}$ are in a finite field F_p with p elements
- D computes all shares: $s_i = f(i)(\text{mod } p)$ for i =1, . . . , n
- Then, D outputs a list of n shares $(s_1,s_2, \ldots ,s_n)$ and distributes each share s_i to corresponding shareholder Pi privately

Share Reconstruction Algorithm:

This algorithm takes any t shares $(s_1,s_2, \ldots ,s_n)$ as input, it can reconstruct the secret s as

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

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

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

In Shamir's (t, n)-SS, the secret of each shareholder is just the y-coordinate of f(x) and the x-coordinate is made publicly known.

III PROPOSED MODEL

In this paper, we propose a solution based on the approach using any (t, n) secret sharing scheme to distribute a group key to a group consisting of (t – 1) members and provide confidentiality and authentication for distributing group keys. SS-scheme consists of three steps:

Initialization of KGC

The KGC randomly chooses two safe primes p and q 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 U_i, where $x_i, y_i \in Z_n$.

Group Key Generation And Distribution

This process involves five steps:

Step 1: The initiator sends a key generation request to KGC with a list of group members as $\{U_1, U_2, \ldots, U_t\}$

Step 2: KGC broadcasts the list of all participating members, $\{U_1, U_2, \ldots, U_t\}$, as a response

Step 3: Each participating group member needs to send a random challenge, $R_i \in Zn$ to KGC

Step 4: KGC randomly selects a group key, k and generates generates f(x) ,(0,k) and $(x_i, y_i \oplus R_i)$ for i = 1, . . . , t. KGC also computes t additional points, Pi, for i = 1, . . . , t, on f(x) and Auth = $h(k, U_1, \ldots, U_t, R_1, \ldots, R_t, P_1, \ldots, P_t)$. KGC broadcasts {Auth, P_i}, for i = 1, . . . , t, to all group members

Step 5: Each group member, U_i, knows the shared secret, $(x_i, y_i \oplus R_i)$, and t additional public points, P_i on f(x), is able to compute f(x) and recover the group key k = f(0). Then U_i computes $h(k, U_1, \ldots, U_t, R_1, \ldots, R_t, P_1, \ldots, P_t)$ and checks with Auth. If these two values are identical, U_i authenticates that the group key is sent from KGC.

IV SECURITY ANALYSIS

There are two types of adversaries in our proposed protocol, insider and outsider. 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.

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. There are three security goals for our key transfer protocol:

Key freshness

Random group key selected by KGC to recover key is a function of random challenge selected by each group member.

Key confidentiality

Authorized user only knows t+1 points and is able to reconstruct the polynomial f(x) and recover the group key k. Unauthorized user knows only t points on f(x) available. Thus, unauthorized member knows nothing about the group key.

Key authentication

Since the group key is known only to authorized group members and KGC, unauthorized members cannot forge Auth value. 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 KGC.

Remarks:

- In our proposed protocol, we focus on protecting group key information broadcasted from KGC to all group members
- User authentication can be achieved based on the knowledge of the shared secret between each user and KGC. User's challenge to KGC can be authenticated by KGC if each user attaches an authentication value, along with the challenge message. Furthermore, key confirmation can be done by asking each group member to send a key confirmation. Then, after receiving all key

confirmations from group members, KGC sends a group key confirmation to each group member

V RESULT

Construction

Suppose that our secret S is 1234. We wish to divide the secret into 6 parts (n=6), where any subset of 3 parts (t=3) is sufficient to reconstruct the secret. At random we obtain 2 numbers: 166, 94 (a_1=166; a_2=94). Our polynomial to produce secret shares is therefore: $f(x)=1234+166x+94x^2$

We construct 6 points from the polynomial: (1,1494), (2,1942), (3,2578), (4,3402), (5,4414), (6,5614) and we give each participant a different single point (both x and f(x))

Reconstruction

In order to reconstruct the secret any 3 points will be enough. Let us consider (x0,y0)=(2,1942); (x1,y1)=(4,3402); (x2,y2)=(5,4414). We compute Lagrange basis polynomials:

$$l_0 = \frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2} = \frac{x-4}{2-4}\cdot\frac{x-5}{2-5}$$
$$= \frac{1}{6}x^2 - 1\frac{1}{2}x + 3\frac{1}{3}$$
$$l_1 = \frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2} = \frac{x-2}{4-2}\cdot\frac{x-5}{4-5}$$
$$= -\frac{1}{2}x^2 + 3\frac{1}{2}x - 5$$
$$l_2 = \frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1} = \frac{x-2}{5-2}\cdot\frac{x-4}{5-4}$$
$$= \frac{1}{3}x^2 - 2x + 2\frac{2}{3}$$

Therefore,

$$f(x) = \sum_{j=0}^{2} y_j . l_j(x)$$
$$= 1942.\left(\frac{1}{6}x^2 - 1\frac{1}{2}x + 3\frac{1}{3}\right) + 3402.\left(-\frac{1}{2}x^2 + 3\frac{1}{2}x - 5\right)$$
$$4414.\left(\frac{1}{3}x^2 - 2x + 2\frac{2}{3}\right)$$
$$= 1234 + 166x + 94x^2$$

The secret is the free coefficient, which means that s=1234, and we are done.

Remarks:

- KGC shares a secret (x_i,y_i) with each user, adding or removing each user does not need updation in existing shared secret
- For distributing a secret group key involving t group members, KGC needs to broadcast a message containing (t + 1) elements to all group members
- Each group member needs to compute a t-degree interpolating polynomial f(x) to decrypt the secret group key
- In Shamir's (t, n)-SS, the secret of each shareholder is just the y-coordinate of f(x) and the x-coordinate is made publicly known
- In Shamir's (t, n)-SS, the modulus p used for all computations is a prime number
- In our proposed protocol for security reason, we need to keep both x-coordinate and y-coordinate as each user's secret. Furthermore, to prevent insider attack, the modulus n used for computations is a composite integer

VI CONCLUSION

Every user needs to register at a trusted KGC initially and preshare a secret with KGC. KGC broadcasts group key information to all group members at once. The confidentiality of our group key distribution is information theoretically secure. We provide group key authentication. Security analysis for possible attacks is included. Thus we have proposed an efficient group key transfer protocol based on secret sharing.

But, the proposed protocol is only suitable for distributing secret group key to a group with a small group size. Also that the fault detection is not acknowledged and so the broadcast is not reliable. Moreover there is computational load of each group member.

In order to overcome all these, KGC generates a private key and a sequence of systems during registration. This private key is transmitted but the sequence does not get transmitted. The sequence is closed automatically at random interval. KGC closes the session when malicious user intrudes and it informs the change of sequence. Clients can request KGC for a higher number of sequence for critical data transmission and so the authentication time is more. Based on the clients security level sequence can change and hence proved flexible security.

REFERENCES

[1] IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004
[2] 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.
[3] G.R.Blakley, "Safeguarding Crypto-graphic Keys," Proc. Am. Federation of Information Processing Soc. (AFIPS '79) Nat'l Computer Conf., vol. 48, pp. 313-317, 1979.
[4] A. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, no. 11, pp. 612-613, 1979.
[5] S. Berkovits (1991) "How to Broadcast a Secret," Proc. Eurocrypt '91 Workshop Advances in Cryptology, pp. 536-541
[6] H. Harney, C. Muckenhirn, and T. Rivers, (July 1997) "Group Key Management Protocol (GKMP) Architecture," RFC 2094
[7] C.H. Li and J. Pieprzyk, (1999) "Conference Key Agreement from Secret Sharing," Proc. Fourth Australasian Conf. Information Security and Privacy (ACISP'99), pp. 64-76
[8] G. Saze, "Generation of Key Predistribution Schemes Using Secret Sharing Schemes," Discrete Applied Math., vol. 128, pp. 239-249, 2003.
[9] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM, vol. 21, pp. 120-126, 1978.
[10]J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," J. Cryptology, vol. 20, pp. 85-113, 2007.