Journal of Engineering and Applied Sciences 7 (3): 285-289, 2012
ISSN: 1816-949X

An Optimal and Cost Effective Key Management Scheme for Secure Multicast Communication

J.K. Sridhar, R. Senthil Kumar and S. Arun Kumar
School of Computing, SASTRA University, Thanjavur, India

Abstract: The recent growth, popularity and extensive need for the deployment of group oriented applications like multiparty video conferencing, multiplayer online gaming and real time push based delivery systems has triggered the demand for secure group communication. These applications require multicast to minimize the volume of network traffic they generate. Multiparty communications have recently become the focus of new developments in the area of applications. The goal of this study is to establish a secure group communication using multicast key distribution scheme. In general, to meet forward secrecy and backward secrecy, any change in the group membership will induce group rekeying. Here, a novel key management scheme is proposed in which no redistribution of group key (shared key) is required during the group dynamics there by reducing computation and communication complexities considerably.

Key words: Multicast, group key management, rekeying, network traffic, demand, India

INTRODUCTION

Multicast communication has been anticipated as an effective way to disseminate data to potentially large number of receivers (Wong *et al*., 2000) (from one sender to multiple receivers or from multiple senders to multiple receivers). If the same data is to be sent to different destinations, multicast is preferred to multiple unicast. The advantage of multicast is that it:

- Makes better utilization of bandwidth (saves up to 1/N of the bandwidth compared to N separate unicast clients)
- Enables the desired applications to service many users without overloading a network and resources in the server
- Reduces host/router processing (if same data is sent to multiple receivers)

In multicast networks, users are organized as a group. The popularity of this secure group communication (Moyer *et al*., 1999; Canetti and Pinkas, 1999; Canetti *et al*., 1999; Hardjono *et al*., 1999) is fuelled by the growing importance of group oriented and collaborative applications. The application includes video/audio broadcast (one sender), video conferencing (many senders), real time news distribution, interactive gaming. Security is an important concern for any communication mechanism-multicasting is no exception. For achieving privacy and integrity of the multicast session, group management becomes an important aspect in secure multicasting environment. The fundamental secure challenge in secure group communication revolves around secure and efficient key management owing to group dynamics. Each user holds a group key (a secret quantity say Gk which is shared by current members of the group) and an individual key (say Uk). In order to send a message to the group, the user encrypts the message with the group key. All the members of the group can now decrypt this message to its original form using the group key. Non-members of the group cannot have access to the message since they do not possess a valid group key. An unbiased authentication agent (say key server) administers all these users. When user(s) join(s)/leave(s) the multicast group, the group key has to be changed to comply with forward (to prevent expelled member from deciphering current and future multicast communication) and backward secrecy (to prevent new joined group members from accessing the past multicast communication). There is a overhead in managing the keys as users enter/leave the group/network. To address this issue, researchers devise a key management strategy that eliminates the rekeying process by allowing the participants of the group to compute the group key on every membership change. The key server must keep track of member join/leave operation.

LITERATURE REVIEW

There are various proposed schemes for key management in multicast groups (schemes that use minimal number of keys to complex hybrid tree key

Corresponding Author: J.K. Sridhar, School of Computing, SASTRA University, Thanjavur, India

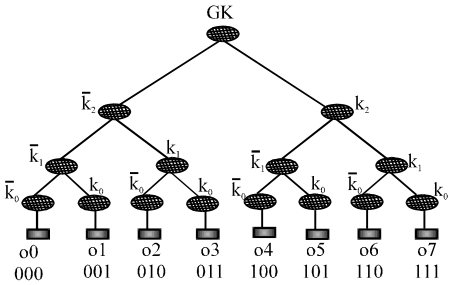

Fig. 1: Key tree

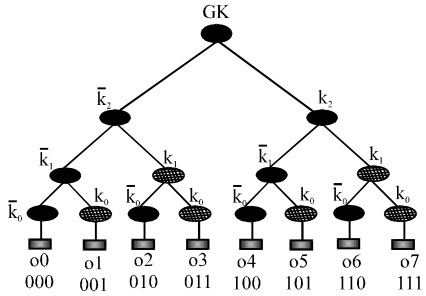

Fig. 2: Multiple member removal

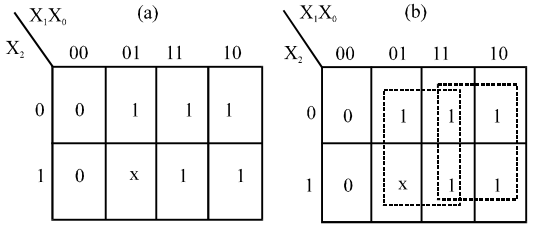

Fig. 3: a, b) Karnaugh map minimization of membership function

distribution scheme). Of these the several techniques, Boolean Minimization Technique (Chang *et al*., 1999) is studied in detail to understand the aspects for incorporating in our proposed technique.

BOOLEAN MINIMIZATION TECHNIQUE

In Boolean minimization technique, every user is assigned to a unique key called UserID (UID). The length of the UID is based on the number of users in the group and is calculated as:

$$\text{Length of UID} = \log_2 N$$

where, N is the number of users in the group. The UID can be represented as $X_{n-1}, X_{n-2},..., X_0$ where X_i can take values either 0 or 1. The members receive the following two different keys in order to participate in the group. Group key are used to decrypt or encrypt data intended for the group members. Auxiliary keys are a set of keys to update the group key in a secure manner. The implementation of the key management scheme employs a key structure. The sample key tree structure constructed by the group controller with eight users is shown in Fig. 1.

Members join and leave operations

Individual member removal: Whenever a member of a multicast group is to be expelled, new group key needs to be disseminated to every member except the one who departed to make sure that the expelled member can no longer send and receive data addressed to the group. In order to update the new group key GK, the controller has to compute the group key GK_{new} and this is encrypted with the complementary of the auxiliary keys of the departed member.

Multiple member removal: In practice, there are number of situations in which many users may leave at a time. Under such situations, there must be a way to provide secure multicasting for only the remaining valid users. The multiple removals of users can be dealt with Boolean Minimization Technique and the same is shown in Fig. 2. Let us consider the same example as shown in Fig. 2 where two members' c0 and c4 are leaving the group. The membership function for the available members is 1 and for the evicted member is 0.

Using the member function, karnaugh map is constructed as shown in Fig. 3. Each field of the karnaugh map corresponds to a specific minterm and is marked as 0, 1 or X (for dummy nodes). The next step of the minimization procedure is to identify the largest possible rectangle that contains 1. These rectangles are called prime implicants of the function and by choosing the minimum number of the prime implicants the minimum SOP of the function is obtained.

For this example, the minimization function is (k0+k1) and the new group key is multicasted with the minimization function. It is evident that the left users' c0 and c4 does not possess either k0 or k1 but all the other users have either k0 or k1 and hence they can decrypt the new group key. The rekeying message now required is only 2 unlike 6 if the leaves are considered separately.

Join: Whenever a new member joins the group the centralized server gives the UID to the new member and calculates the new group key. It is first sent to the new member by unicast. It is then encrypted by the old group key and sent to all the remaining members by one multicast. This can be further enhanced by considering the following three scenarios.

- Number of leave request equal to join request
- Number of leave request is less than join request
- Number of leave request is greater than join request

In existing key distribution schemes (Chang *et al*., 1999; Wong *et al*., 2000), auxiliary keys are employed in forming the group key on a membership change of the group. To update the group key, multiple encryptions of the message by a new session key (formed using auxiliary key) are involved. This study describes a technique in which no supplementary keys are required for generating the group key reducing the keying cost. This is achieved by employing a scheme which allows participants to compute the group key using a random number generated by the key server during group dynamics.

PROPOSED SCHEME FOR KEY MANAGEMENT (MODULO BASED SCHEME)

The proposed modulo based architecture is a star-based group key management scheme for secure multicasting. The proposed scheme differs from the previous research as it does not necessitate in maintaining the key tree architecture. In addition, it eliminates the rekeying process when a group member leaves or joins. The proposed scheme includes two main entities:

Key server: Which is responsible for generating and maintaining the keys for the users as well as the group.

Members: The individuals/users who actively participate in the group communication.

Consider a group with five members (U_1-U_5). The group server allocates a unique number to each member of the group (called as user key (Uk_i)) sticking to the condition that they must be relatively prime to each other. Figure 4 shows the topology of the proposed key management scheme.

Key assignment: If there is only one trusted entity (say key server) that controls the access rights for each group

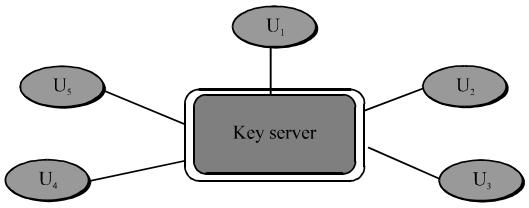

Fig. 4: Star topology of the modulo architecture

member then N members of the group (U_1, U_2, ..., U_N) must be authenticated and identified by the key server. The key server assigns a secret key to every member of the group. The steps involved for the key assignment in the proposed scheme includes:

The key server choose N different relative prime numbers (user key) for each user, i.e., the user U_1 will be assigned with p_1, U_2 with p_2 and so on. All numbers from p_1-p_N should be relatively prime, i.e., GCD (p_i, p_{i+1}) = 1.

The server generates a random number in such a way that which upon dividing by the user key yields the same remainder for all N members of the group, i.e., RN mod p_1 = RN mod p_2..... = RN mod p_N. Using the remainder obtained in the previous step, the group key is calculated using the following equation:

$$\text{Group key} = \left[\sqrt{RN} - \left(REM^X\right)\right] MOD\, P + \varphi(n)$$

Where:

$n = (X - 1) \times (P - 1)$

$\varphi(n) = n - 1$

X, P (a very large prime number) are public values known to all members of the group.

Users entering and leaving the group

User(s) entering the group: Whenever a new member wants to join the group, the key server assigns a unique user key to that member. To achieve forward and backward secrecy, it is essential to change the group key whenever a new member join or the existing member leave off the group.

Key generation process (when user join(s)): The process includes the following steps to be carried for managing the key:

- The server assigns a unique user key to each registered user of the group
- The server generates a Random Number (say RN) which yields the same remainder upon division by the user key including the new member's user key and communicates the same through a secured channel to every member of the group
- Using the remainder (that were the same for all the users), the group key is calculated by the participant for deciphering the message using the formula discussed

For example, there are five members (U_1,..., U_5) in the group. The user keys (Uk) assigned by the server to the members are 3, 4, 5, 7 and 11, respectively (Note that the

user keys are relatively prime to each other). The server generates a random number using the equation:

$$RN = M \times (Uk_1 \times Uk_2 \times ... Uk_5) + \text{remainder}$$

where, M≥1. Let M = 1 and remainder be 2. The generated RN is 4622 (Since, 1× [3×4×5×7×11] +2 = 4622). The Random Number (RN) satisfies the condition that the remainder obtained on dividing each user key by RN is same for all users.

The group key is calculated using the equation stated in this study (taking X = 3, P = 5 [normally P is a very large prime number]):

$$\begin{aligned} \text{Group key} &= \left[\sqrt{4622} - \left(2^3\right)\right] \text{MOD} 5 + \varphi(n) \\ &\quad [68-8] \text{MOD} 5 + \varphi(8), \\ &\quad \left[\text{Since}, n = (X-1) \times (P-1)\right] \\ &= 60 \text{MOD} 5 + 7 \left[\text{Since}, \varphi(n) = n - 1\right] = 7 \end{aligned}$$

User(s) leaving off the group: Whenever a member of a multicast group leaves, the group key that was in use must be updated (in order to achieve forward secrecy). Generally, leaving process is much complex when compared to the joining process due to the possession of the old group key by the leaving member.

Key generation process (when user leave(s)): The process includes the following steps to be carried for managing the key:

The server generates the random number in such a way that when dividing this random number by user key of all the members would result in the same remainder for members who are still in the group (excluding the departing member) and communicates to every member of the multicast group through a secured channel. The group key calculation is same as the joining process.

Say for example, U_4 and U_5 are leaving the group. The server generates the random number 62 and multicast the number to every member in the group. The random number satisfies the condition, i.e., dividing 62 by 3-5 results in remainder 2 (7 and 11 yields other than 2 as remainder). The message is transmitted by encrypting the message with the group key. The members of the group obtain the original message by decrypting the cipher message.

The proposed scheme ensures that only the members of the group access the original message (even though, the non-members of the group have Cipher information, they cannot get original message due to the possession of improper group key).

PERFORMANCE COMPARISON

Experiments were simulated for the various cases discussed before and the effectiveness of the proposed key management scheme is evaluated based on the following metrics:

- Storage complexity
- Communication complexity
- Computation complexity

Storage complexity can be calculated by the amount of space required to store all the keys that were used for multicast communication. It includes the storage requirements of both service provider and users. In modulo arithmetic key management, the server has to store all the user keys with one group key and two public values (random values -X and prime number P used for calculating the group key) and each member have to store its own private key plus group key. If there are N member in the group then the server stores N+3 and each member stores a user key and a group key. So, the overall complexity is N+5.

Communication complexity is measured in terms of number of rekeying messages sent by the service provider and computation complexity by the number of encryptions needed by the service provider. Both complexity metrics depend upon the position of the existing members in the tree after the left out members. In the proposed model, the communication cost is O(1) for join and leave operation, i.e., one message is adequate for the rekeying process (computation of group key) and it does not depend upon the number of

Table 1: Comparison of the complexities of various key management schemes

Parameters	DEP	Key graph	HBT	Boolean minimization	Modulo based
No. of keys in the multicast group	n+i+1+c O(n)	n(1+log n) O(n)	2 log n+1 O(log n)	2 log n+1 O(n)	N+5
No. of keys managed by the sender	c+2	(d+2)(h - 1)/2	2 log n+1	2 log n+1	N+3
No. of keys at a member	4	d/(d - 1)	O(log n)	log n+1	2
No. of message at join	O(1)	O(1)	O(log n)	Not dealt	O(1)
No. of message at leave	O(1)	O(log n)	O(log n)	<log n	O(1)
Total key encryption during data transmission	O(1+c)	O(1)	O(1)	O(1)	O(1)
No. of key encryption at the sender	O(c)	O(1)	O(1)	O(1)	O(1)

members leave/join the group. The complexity of various key management schemes are tabulated (Table 1) to show the reduction in overhead (for keying) in the proposed scheme.

CONCLUSION

Modulo based key management scheme was developed by studying the security issues (group membership, key management processes and scalability) concerned with secure multicast communication. The proposed strategy has a number of notable features:

- The star based architecture of the proposed scheme facilitates a way to reduce the complexity beyond log (n)
- The proposed scheme does not require auxiliary keys for computing the group key when member join(s)/leave(s) off the group
- It is based on de-centralized architecture

Besides the key server, the members of the group also actively participate in forming the group key. Hence, the load on the server is much reduced. The proposed scheme also satisfies the cryptographic property namely, group key secrecy which ensures that for anyone who is not the member of existing group, it is computationally infeasible to calculate the group key. This results in a key management scheme that provides an efficient framework for secure group communication at minimal cost. This research can further be extended to reduce the burden placed on the key server for key distribution.

REFERENCES

Canetti, R. and B. Pinkas, 1999. A taxonomy of multicast security issues. IETF Internet Draft (Work in Progress), April 1999.

Canetti, R., P.C. Cheng, D. Pendarakis, J.R. Rao, P. Rohatgi and D. Saha, 1999. An architecture for secure internet multicast. http://www.securemulticast.org/smug2-rohatgi.pdf.

Chang, I., R. Engel, D. Kandlur, D. Pendarakis and D. Saha, 1999. Key management for secure internet multicast using boolean function minimization techniques. Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, March 21-25, 1999, New York, USA., pp: 689-98.

Hardjono, T., R. Canetti, M. Baugher and P. Dinsmore, 1999. Secure IP multicast: Problem areas, framework and building blocks. http://www.securemulticast.org/msec-bof-4-Baugher-FrameworkPDF.PDF.

Moyer, M.J., J.R. Rao and P. Rohatgi, 1999. A survey of security issues in multicast communications. IEEE Network, 13: 12-23.

Wong, C.K., M. Gouda and S.S. Lam, 2000. Secure group communication using key graphs. IEEE/ACM Trans. Networking, 8: 16-30.