A Conventional Authentication in Key Management using Progressive Approach

Sandosh Sakkarapany[1], Uthayashangar Sundaramourty[2]
[1]Assistant Professor
Department of Information Technology,
Manakula Vinayagar Institute of Technology,
Puducherry-605 107,India.

[2]Assistant Professor
Department of Information Technology,
Manakula Vinayagar Institute of Technology,
Puducherry-605 107,India.

Abstract

Secure and reliable group communication is an increasingly active research area prompted by the growing popularity of many types of group-oriented applications. The main building block to achieve security in group communication scenarios is management of the secret information that should be known only to group members involved in communication. However, the complete security with the key management for such protocols remains a significant problem. In this paper, a secure group key agreement between the group members is proposed. The main advantage of the proposed system is the continuous improvement in Authentication between the group key members.

Keywords*: Multicast, Cryptography, TGDH, Group Key management, Continuous Authentication.*

1. Introduction

Services such as Stocks publishing, News distribution, Audio / Video conference systems, Collaborative work-flow systems and any software updates, use multicasting information among many people. However, an increasing number of such applications require secure multicast services in order to control the Group Members in a secure way. Group Key Management is a methodology which helps to achieve security in multicasting information over group-oriented communication.

In order to multicast information among a certain group securely, a group key should be shared among all members in the group. Every information packages should be encrypted with the shared group key before they are transmitted. Only the authorized users who have the shared common group key can decrypt the package and retrieve the information. The unauthorized users perhaps received the encrypted packages; they cannot retrieve the information without the group key.

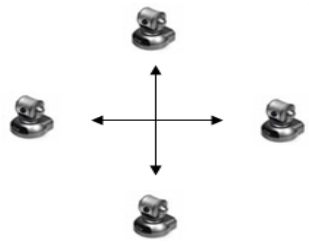

Fig. 1 Many to Many Group Communication

But there are some problems need to be addressed in this way of sharing information. The members in the group may be changed frequently. Any time when a new member joined, a new group key should be generated and distributed to all group members, include the new joined member. This rekeying (generation and distribution of new key) process helps to maintain continuous Authentication to share confidential information among the authorized users.

2. Key Management

Multicasting helps the Group Member to send the confidential information to the selected group of recipients in a group. The Key Management can be classified along two primary directions: group key distribution (transport), and group key agreement.

The main difference is that in case of group key distribution there exists a designated participant with extended rights, called group manager or key server that computes the group key on its own and distributes it securely to all other participants, whereas in case of

group key agreement all participants have equal rights and each of them provides own contribution to the computation of the group key.

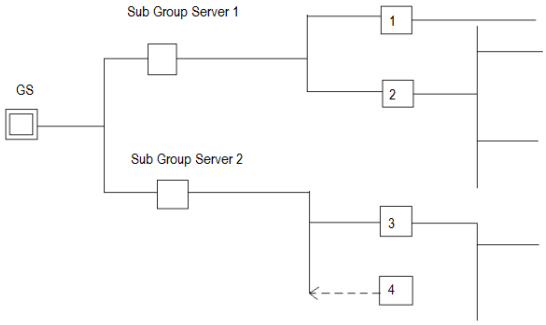

Fig. 2 Key Management

In a Group-Oriented system there is Group Server or Key Distributor, who allows the group members in the group to access data. Initially when any user wants to join a group, the user should request the Group Server that the user is interested in joining the group.

Then the Group Server asks for the one time authentication from the new user which is nothing but Username and Password. After this process the Group Server introduces two keys to the new agent which is called the Group Key and the Individual Key. The Group Key is used to decrypt the confidential information shared among the group members.

All the members in the group have the same Group Key where as the Individual Key is unique and is used to request the Group Server that the particular user in the group is interested in leaving the group. Each Group Server or Key Distributor generates and distributes the Group Key. Then the Group Server encrypts the confidential information with the Group Key and send it to the other member in the group. The group members receive and decrypt the message with the shared Group Key distributed by the Group Server.

Key refreshing is one of the most important security requirements of group key agreement protocols. Because, the session key should be known only to the involved parties.

The following four properties should be guaranteed:

2.1 Group Key Secrecy-It is computationally infeasible for a passive group member who quit the group to discover any group key.

2.2 Forward Secrecy -The new group member who knows a continuous subset of group keys cannot discover any old group key to decrypt the messages.

2.3 Backward Secrecy - A passive group member who knows a continuous subset group keys cannot decrypt the shared information anymore with the key.

2.4 Key Independence -The Individual Key shared between the group server and particular group member is independent.

3. TGDH

From all existing group key agreement protocols, the approach called Tree-Based Group Diffie-Hellman (TGDH) protocol suite is chosen for better efficiency. These protocols combine the structure of binary key trees with two-party Diffie-Hellman key exchange protocol to achieve the computation of the secret group key after several rounds. This method is also called an iterative Diffie-Hellman (IDH) key exchange.

This solution is secure, surprisingly simple and also very efficient, compared to other existing group key agreement protocols. TGDH protocol suite is most applicable in dynamic groups, where number of participants changes during the communication period.A group key agreement scheme needs to provide key adjustment protocols stemming from membership changes. TGDH suite includes protocols supporting the following operations:

3.1 Leaving the group -If any member leaves the group rekeying (generation and distribution of new Group Key will be done by the Group Key Server. When a user leaves the group, the Group Key Server generates and distributes the new Group Key.

Fig. 3 Leaving the TGDH tree

3.2 Joining the Group-When a new user enters the group the Group Key Server distributes the new Group Key to the entire users including the new one.

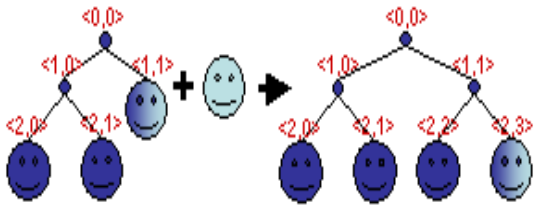

Fig. 4 Joining the TGDH tree

Initially the Group Server is created, then the new group members request the Group Server to enter into the group, then the one time authentication is to be done for each new user successfully. Once the process is done the TGDH Protocol suite forms the group in a binary tree structure for managing the group member in an efficient manner.

4. Skipjack

Skipjack is the cryptographic algorithm which encrypts and decrypts data in 64-bit blocks, using an 80-bit key. It takes a 64-bit block of plaintext as input and outputs a 64-bit block of ciphertext. Skipjack has 32 rounds, meaning the main algorithm is repeated 32 times to produce the ciphertext. It has been found that the number of rounds is exponentially proportional to the amount of time required to find a key using a brute-force attack. So as the number of rounds increases, the security of the algorithm increases exponentially.

This cryptographic algorithm used for encrypting and decrypting the confidential information brings more security on the information shared among the group member. Now the information is more secure and the users who access the information are legitimate is to be addressed.

Even though multicasting information over the group oriented environment with the TGDH key management protocol suite, there are still some problems need to be addressed.

Multicast protocol would be subject to attack by an **active intruder** compared to a unicast protocol. There are inherently more opportunities for interception of traffic, would typically make it easier for an intruder to pose as another legitimate principal.

5. Proposal

The important problem need to be addressed here is Authentication. Because multicast protocols are easily attacked by active intruders and interception is possible. In order to prevent intruders and maintaining security, a step by step continuous improvement in authentication is required.A multicast protocol is easily subject to attack by an intruder compared to unicast protocol and there are inherently more opportunities for interception.

In the Group Key Server asks for one time authentication (Username and Password) then it generates a new Group Key and Individual Key to the users in the group. After the one time authentication the Group Key Server does not worry about whether the confidential information reaches the authorized destination or user.

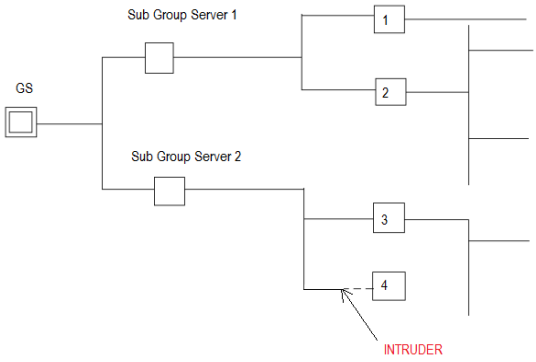

Fig. 5 Intruder

Hence any intruder may receive the encrypted messages from the Group Key Server. Since the intruder do not have the group key he cannot decrypt the encrypted information shared among the group members. But, if any of the users leaves or joins the group, rekeying will be done by the Group Key Server and the intruder will get the new group key then the intruder starts decrypting the confidential information which he has not been authorized to read.

To avoid intruders into the Group, accessing the confidential information, frequent authentication verification is to be done. It is the role of the Group Server keep tracking the information send by the Group Server reaches only the legitimate user. A separate process of cryptography is used to confirm that the users in the Group are original or legitimate user.

5.1 Conventional Authentication

Once the one time authentication (requesting the Username and Password) is done the Group Server generates and distributes new Group Key to all the members of the Group and an individual key for the new user. The Group Key is used to decrypt the confidential encrypted messages by all the members and the

individual key is used to inform the Group Server when the member is interested in leaving the Group.
In the proposal, it has been planned that a tiny bit of plain text other than encrypted confidential Group message is sent to all the members in the Group at regular intervals of time. Each member in the Group receives the tiny bit plain text and encrypt it with the unique Individual Key and sends the result of encryption (ie) cipher text back to the Group Server. Then the Group Server also calculates cipher text for that plain text separately in the Server side and compares the resultant value with the received cipher text from each member.

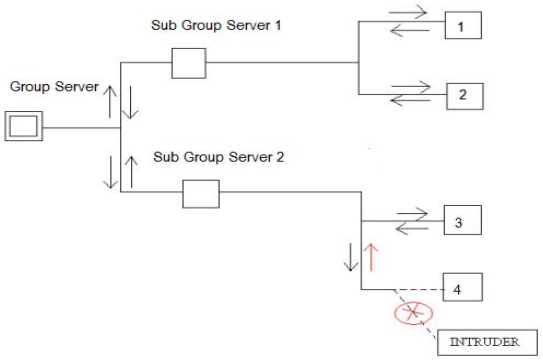

Fig. 6 Conventional Authentication

Any intruder who enters the communication link between the member and the Group Server may get the group key if any one of the member joins or leaves the Group, but the intruder is not possible to get the unique individual key.

Only legitimate group member who is having the unique Individual Key can provide the expected results of cipher text to the server. After receiving the result of encryption from each member the server compares the result.

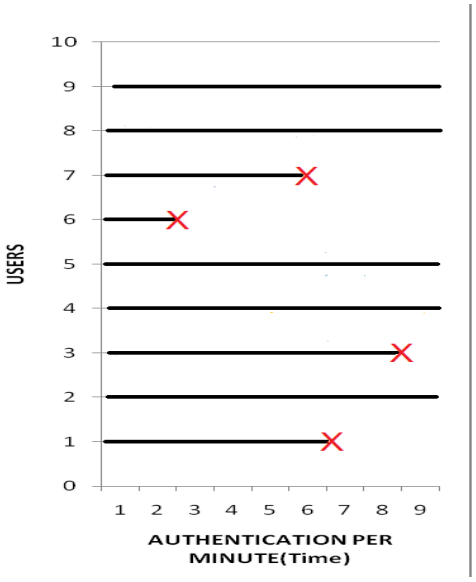

Fig. 7 Step by Step improvement

If match is found with the encrypted result of particular user, the communication continues.

If the result received by the server is identified as wrong one, the server comes to know some unauthorized person is trying to access the information then it stops the communications with the particular group member.

6. Conclusion

Sharing confidential information among the Group Members in a secure way is more critical. After the one-time authentication, the new Group Member enters the group and starts sharing the confidential information by Encrypting and Decrypting the messages with the help of the Group Key provided. Skipjack- an efficient Encryption Algorithm helps to maintain the information more secure among the Group Members. During the process of sharing of the encrypted information among the members, it is not assured that the information reaches only the legitimate Group Members. After the one-time Authentication (Username and Password) it is possible for the Intruders to enter the group and receives the confidential information but, the Intruder cannot decrypt the encrypted information without the group key.

In Group Key Management, if any member joins or leaves the group rekeying (generation and distribution of new group key) is done. Hence the Intruder easily receives the Group Key and decrypts the confidential information. Once the new member joins the group, the Group Members shares the information without worrying

about whether the confidential message reaches the legitimate Group Members or not.

With the proposed system, Conventional Authentication is introduced and maintained till the Group Member quits the group. If any Intruder has entered the group, the intruder cannot perform the Authentication Verification process with the Group Server successfully. If the Authentication Verification Process fails, the communication between the Group Server with the intruder is terminated. Hence, the confidential information among the Group Members becomes more secure through conventional Authentication.

7. References

[1] Feng Zhu, Wei Zhu, Matt W. Mutka, "Private and Secure Service Discovery via Progressive and Probabilistic Exposure" VOL. 18, NO. 11, IEEE November 2008.

[2] Ling Cheung, Joseph A. Cooley, Roger Khazan, Calvin Newport, "Collusion-Resistant Group Key Management Using Attribute-Based Encryption" MIT Lincoln Laboratory, March 22, 2007.

[3] W. Yu, Y. Sun, and K. J. R. Liu, "Optimizing Rekeying Cost for Contributory Group Key Agreement Schemes," IEEE Transactions On Dependable And Secure Computing, vol. 4, no. 3, pp.228-242, 2007.

[4] F. Zhu, M. Mutka, and L. Ni, "A Private, Secure and User-CentricInformation Exposure Model for Service Discovery Protocols," Mobile Computing, vol. 5, pp. 418-429, IEEE 2006.

[5] F. Zhu, M. Mutka, and L. Ni, "Service Discovery in Pervasive Computing Environments," Pervasive Computing, vol. 4, pp. 81-90, IEEE Oct 2005.

[6] Rahul.S and Hansdah, RC , "An Efficient Distributed Group Key Management Algorithm" Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004, 7-9 July, California, 230 -237.

S.Sandosh received the B.E degree in Computer Science and Engineering from Anna University, India in 2007 and M.Tech degree in Information Security from Pondicherry University, India in 2010. He is working in Manakula Vinayagar Institute of Technology,Puducherry, India as a Assistant Professor in Information Technology Department.

S.Uthayashangar received the B.Tech degree in Information Technology from Pondicherry University, India in 2007 and M.Tech degree in Information Security from Pondicherry University,India in 2010. He is working in Manakula Vinayagar Institute of Technology,Puducherry,India as a Assistant Professor in Information Technology Department.