The Software Architecture of A Secure and Efficient Group Key Agreement Protocol

Noe Lopez-Benitez[1*]

[1]**Department of Computer Science, Texas Tech University**

Abstract Group communications are becoming popular in Internet applications such as video conferences, on-line chatting programs, games, and gambling. Secure and efficient group communication is needed for message integration, confidentiality, and system usability. However, the conventional group key agreement protocols are too much focused on minimizing the computational overhead by concentrating on generating the common group key efficiently for secure communication. As a result, the common group key is generated efficiently but a failure in authentication allows adversaries to obtain valuable information during the group communication. After achieving the secure group communication, the secure group communication should generate the group key efficiently and distribute it to group members securely, so the balance of security and system usage must be considered at the same time. Therefore, this research proposes the software architecture model of a secure and efficient group communication that will be imbedded into networking applications.

Key Words : group key agreement protocol, software architecture model, network security

1. INTRODUCTION

There are the two facets in this research. One is to improve security and the other is to enhance the efficiency of the group communication algorithm. Security and efficiency are critical issues in group communications [1] and present significant key management challenges because both security and efficiency must be considered at the same time. Group Key (GK) plays an important role in secure group communications. In order to achieve and maintain secure communication, the Group Key Agreement Protocol (GKAP) [2] requires that all members in the group contribute to the generation of the GK and use the GK to encrypt and decrypt messages over insecure networks.

The function for generating GK in the group key agreement is a modular exponentiation. In Tree-based Group Diffie-Hellman (TGDH), the modular exponentiation is the most expensive computation operation [3]. The number of exponentiations for membership events depends on the number of group members.

TGDH assumes all users have equal computing power. The overall computing power is estimated by hardware and software, such as CPU specifications, CPU usages, memory size, input/output bandwidth, and communication latency [4]. Unless each user's computing power is considered in assigning the group member sequence, all members can be negatively affected by the duration of the group key generation. Therefore, each time the group membership changes maintaining a perfectly balanced tree is an efficiency issue in the group key generation process.

Received 2014-08-06 Revised 2014-08-20 Accepted 2014-08-27

[*]Corresponding author: Noe Lopez-Benitez (noe@ttu.edu)

In distributed computing environments computing power is heterogeneous where users could be at workstations, laptops, or mobile computers. The last user has to compute the largest number, k from the cardinal value, k = $\alpha^{x1x2...xn}$mod q

2. GROUP COMMUNICATION

There are two kinds of group communications, one-to-many and peer-to-peer. One-to-many is client server based communication, for example, TV or radio broadcasting, Geographic Position System (GPS), and so forth. In peer-to-peer communication the group size is relatively small, less than 100 and there is no centralized controller [6]. In this research, the term group refers to peer-to-peer group communication. Membership in a dynamic peer-to-peer group communication tends to change frequently. Networks are generally regarded as insecure because they are connected to each other and there is no central controller. A secure communication channel must be established in group communications to protect messages over an insecure network environment. Currently, group key management is being used for establishing a secure communication channel [7].

2.1 Group Key Management

Group communication arises in many different settings: from low-level network multicasting to conferences and other groupware applications. In particular, group communication is often crucial in the battlefield. Regardless of the environment, security services are necessary to provide communication privacy and integrity. For secure communication, group members need a common group key to protect their messages while they are communicating to others. In this context, group key management is responsible for generating the GK and distributing it to each member securely over an insecure network environment, making key management the building block in group key management [8]. Unless the communication channel is secure, delivery of messages over the network to the right destination can not be guaranteed. Group key management is used for establishing a secure channel [14][15].

There are two types of schemes in group key management, group key distribution and group key agreement [9]. In group key distribution one member is designated as the key distribution center. He/she computes the GK and distributes it to each member in the group. This scheme is suitable for client-server environments like multicast [10]. However, peer-to-peer group communication needs a different key generation and key distribution due to the characteristics of peer-to-peer group communication such as dynamic, relatively small number of group size, network partition, and merge [11][12][13].

3. Proposed Software Architecture model for Group Key Agreement Protocol

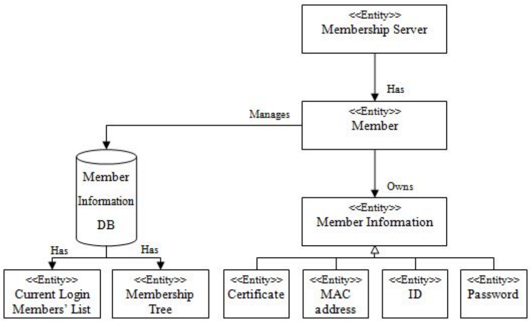

Fig. 1. Group Member's Entity Static Model

Figure 1 shows a group member's entity static model. Each member has the information that is an ID (Identification), a password, a MAC address, and a certificate. Each member may login to the communication program first and then he / she may communicate with others. All members have all members' information that notifies member's status to all other members so that each member will be able to update the current login member list.

Figure 2 describes an overall group key agreement protocol. A new member joins the group, he / she broadcasts his blind key to all members in order to update the membership tree structure and all members participant in generating a common group key for encryption and decryption messages over the network.

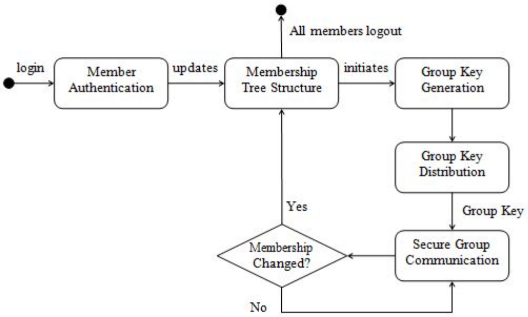

Fig. 2. Group Key Agreement State Chart

There are two ways to update the membership tree structure. One is a decentralized broker synchronization and another is a centralized broker synchronization. Figure 3 shows a decentralized broker synchronization.

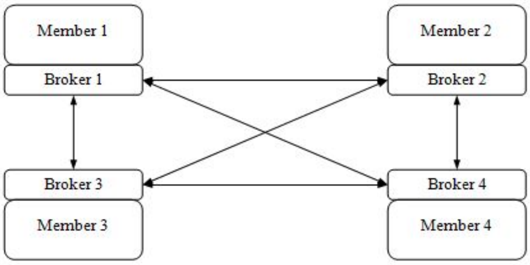

Fig. 3. Decentralized Broker Synchronization

A decentralized broker synchronization is that each member has a broker. Whenever membership changes (join and leave), the broker broadcasts member's status to all other brokers so that all members know the key generation tree structure. If a new member joins the group communication, then a new member's broker broadcasts his / her blind key to all other brokers in order to generate the group key. When an old member leaves, then the member's broker broadcasts his / her leaving to all other brokers so that each member updates the key generation tree and generates new group key. The advantage is that each member knows other's status after membership changes. However, communicational overheads happen.

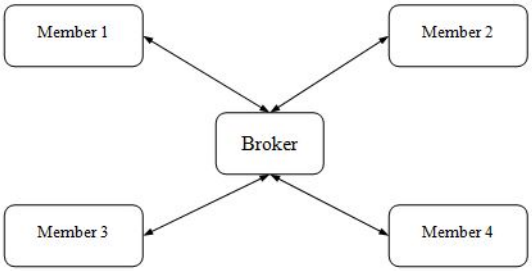

Fig. 4.Centralized Broker Synchronization

Figure 4 shows a centralized broker synchronization. Each member in a centralized broker synchronization can notify his / her status (join or leave) to the broker that broadcasts membership information to all other members. The advantage is less communicational overheads compared to a decentralized broker synchronization. However, it too much depends on broker. For example, if the malfunction of the broker happens, group communication will be stopped.

4. CONCLUSIONS

The basic concept of MGDH is that a user's physical address can be traced in a distributed computing environment. The current CA-based authentication has a weakness. In case a malicious user impersonates a legal member in the group, then other members can make sure whether the certificate belongs to him/her by checking a certificate authority's public key and the public key which is encrypted by the certificate authority's private key. That is a digital signature. However, the users cannot ensure that the name on the public key pair is really a true member's name. The member authentication method in group communication request a support mechanism to compensate for this weakness. Therefore, the user's physical address helps to prevent a fake certificate that has been issued by a certificate authority.

The MAC address-based protocol supports the notion of member authentication by not focusing on who you are, but where you are. In the beginning of the group construction stage, each party does not trust each other, no matter what kind of keys or certificates they have. The MAC address provides a trustworthy relationship because the MAC is a physical address that actual data can reach. Even though the user changes the MAC address, he/she must register the new address with the router in order to communicate with the external world, thereby announcing the physical address. Hence, under all circumstances the group controller is aware of the address of all its members and fraudulent member's Early Bird Attack (EBA) in the group communication can be avoided. Therefore, this approach contributes to establish a secure communication channel.

Security is all about reducing risk, but not eliminating it. There is not a perfect system in the present and no perfect system will exist in the future. Every security effort only makes it harder for adversaries to break into the system. Group communication needs a secure communication channel to prevent eavesdropping on messages. In spite of using a group key, there is nothing that inspires trust in the beginning of communication. Every group member must agree to trust one thing – a trusted third party – and then finally the trust relationship will grow and expand. In the meantime, if an adversary joins the group and pretends to be a legitimate group member at the beginning of the communication stage, there is no way to prevent an Early Bird Attack (EBA). The use of a MAC address is proposed as a security deposit in the beginning of communication stage and it contributes to secure group member authentication. Therefore, a potential adversary might hesitate to join the group if his or her originating physical address is revealed. As a result, a secure user authentication process can be guaranteed when the MAC address is used. After achieving a secure group communication, the group key agreement should generate the GK efficiently and distribute it to group members securely, so the balance of security and system usage must be considered at the same time. Determining computing power must be considered by the elapsed time for calculating GK and communication latency. The distributed computations must consider the variety of members, otherwise the system usage will be degraded. Currently mobile computers are getting more popular, network clusters are communicating with conventional servers, and conventional group key agreements do not consider members' computing power. Therefore, the group key agreement protocol needs to reorder the sequence of members in a key generation tree each time the membership changes based on each member's computing power in order to improve the efficiency of the GK generation.

According to the assumption that a member's computing power significantly affects overall GK generation time, the proposed algorithm has the function to measure the elapsed time for communication cost and calculating the keys in order to determine fast members who will continue to participate in the next level of the GK processes. It is expected that checking computing power for each member would introduce additional overhead. However, the overhead, which comes from the lack of considering user diversity in a group key agreement protocol, will be greater than the overhead associated with determining computing power. We are in the process of comparing the proposed algorithm to other algorithms in regard to all membership events. More members will increase overhead. Thus, we will figure out the maximum membership size that achieves the most efficient GK generation. We will prove that the new group key generation algorithm will be more efficient than current group key agreements by using experimental results. Therefore, reordering will then give members a proper role in generating a GK and as a result, the new algorithm will contribute to maximize the efficiency of the GK generation process.

Finally, MAC-based user identification scheme and reordering user's sequence in the group key generation tree will be able to contribute in establishing a secure and efficient group key agreement.

References

[1] Y. Kim, A. Perrig, and G. Tsudik, "Communication-efficient group key agreement", In 17th International Information Security Conference (IFIP SEC'01), June 2001.

[2] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement", ACM Transaction on Information and System Security, 2004.

[3] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs", IEEE / ACM Transactions on Networking, vol. 8, no. 1, Feb. 2000.

[4]W. Diffie and M. E. Hellman. "New directions in cryptography", Transactions on Information Theory, IT-vol. 22, no. 6, pp. 644-654. Nov. 1976.

[5] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups", In S. Jajodia, editor, 7th ACM Conference on Computer and Communications Security, ACM Press, Athens, Greece, pp. 235 - 244, Nov. 2000.

[6] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architecture", Internet-Draft draft-wallner-keyarch-00.txt, June 1997.

[7] Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, and J. Stanton, "Secure group communication using robust contributory key agreement", IEEE Transaction on parallel and distributed systems, vol. 15, no.4, April 2004.

[8] M. Steiner, G. Tsudik and M. Waidner, "Key agreement in dynamic peer groups", IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 8, pp.769-780, Aug. 2000.

[9] Y. Kim, "Group key agreement: theory and practice", Ph.D. thesis, May 2002.

[10] E. Bresson, O. Chevassut, D. Pointcheval, amd J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange", Conference on Computer and Communications Security Proceedings of the 8th ACM conference on Computer and Communications Security, Philadelphia, PA, pp. 255-264, 2001.

[11] M. Steiner, G. Tsudik, and M. Waidner, "Cliques: A new approach to group key agreement", IEEE ICDCS'98 , May 1998.

[12] A. Fekete, N. Lynch, and A. Shvartsman, "Specifying and using a partionable group communication service", ACM Transactions on Computer Systems, vol. 19, no. 2, May 2001.

[13] Y. Amir,; Y. Kim, and C. Nita-Rotaru, "On the performance of group key agreement protocols", ACM transactions on information and system security, vol. 7, no. 3, p. 457, 2004.

[14] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. http://www.cryptosavvy.com/, Nov. 1999. Shorter version of the report appeared in the proceedings of the Public Key Cryptography Conference (PKC2000) and in the Autumn '99 PricewaterhouseCoopers CCE newsletter. To appear in Journal of Cryptology.

[15] Lan Foster, The Grid: Blueprint for a New Computing Infrastructure, Second Edition, Elsevier, pp.47-53, 2004

저 자 소 개

Noe Lopez-Benitez [Member]

- Ph.D., Purdue University, (1989)
- M.S., University of Kentucky, (1980)
- B.S., Universidad de Guadalajara, (1975)

<Research Areas> : Networking protocol, Security, Distributed computing