A New Privacy Preserving Protocol for Authenticated Group Key Transfer of Data Based on Secret Sharing

Appikatla Srihari [#1], Dr.Koduganti Venkata Rao [*2]

M.Tech Scholar [#1], Professor & HOD [*2]

Department of Computer Science & Engineering,
Vignan Institute of Information Technology,
Visakhapatnam, AP, India.

.

Abstract

In recent days, KGC plays a very important role in generating keys for the user account access in order to provide high level of security. Key transfer protocols rely on a mutually trusted key generation center (KGC) to select session keys and transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration. In this paper, we mainly proposed 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 cannot recover the group key. The confidentiality of this data transfer is always secure. We also provide authentication for transporting this group key. By conducting several experiments on this proposed model , we finally came to an conclusion that with this mechanism we are able to give high security for data transfer as well as bestly suited for reducing key sizes in the data base.

Keywords

Group Key Transfer Protocol, Session Key, Secret Sharing, Confidentiality, Authentication.

1. Introduction

Wireless networks rely on the uninterrupted availability of the wireless medium to interconnect participating nodes. However, the open nature of this medium leaves it vulnerable to multiple security threats. Anyone with a transceiver can eavesdrop on wireless transmissions, inject spurious messages, or jam legitimate ones. While eavesdropping and message injection can be prevented using cryptographic methods, jamming attacks are much harder to counter. They have been shown to actualize severe Denial-of-Service (DoS) attacks against wireless networks

Applications such as conferencing, distributed interactive simulations, networked gaming, and news dissemination are group-oriented. In these applications, it is necessary to secure the group communication as the data are sensitive or it requires the users to pay for it. In the algorithms for secure group communication (e.g., [1], [2], [3], [4]) a group key is shared by all the users. The group key is used to encrypt data transmitted to the group. The group membership is dynamic. When group membership changes, to protect the confidentiality of the current users, a new group key needs to be shared by the users.

In most secure communication, the following two security functions are commonly considered:

Message Confidentiality: Message confidentiality ensures the sender that the message can be read only by an intended receiver.

Message Authentication: Message authentication ensures the receiver that the message was sent by a specified sender and the message was not altered en route.

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. The key establishment protocol also needs to provide confidentiality and authentication for session keys. According to **[5],** there are two types of key establishment protocols: key transfer protocols and key agreement protocols. Key transfer protocols rely on a mutually trusted key generation center (KGC) to select session keys and then transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration. 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 [7]. In DH protocol, the session key is determined by exchanging public keys of two communication entities. Since the public key itself does not provide any authentication, a digital signature can be attached to the public key to provide authentication. However, DH public key distribution algorithm can only provide session key for two entities; not for a group more than two members.

When a secure communication involves more than two entities, a group key is needed for all group members. Most well-known group key management protocols can be classified into two categories:

Centralized Group Key Management Protocols: a group key generation center is engaged in managing the entire group.

Distributed Group Key Management Protocols: there is no explicit group key distribution center, and each group member can contribute to the key generation and distribution.

2. Background Knowledge

In this section we will describe the assumptions and background knowledge that is used for developing the new privacy preserving tool for secure data communication.

2.1 Main Motivation

There are other distributed group key management protocols based on non-DH key agreement approach**.** Tzeng proposed a conference key agreement protocol based on discrete logarithm (DL) assumption with fault tolerance in recent years. The protocol can establish a conference key even if there are several malicious participants among the conference participants. However, the protocol requires each participant to create nn-power polynomials, where n is the number of participants; this is a serious encumbrance to efficiency. In 2008, Cheng and Laih **[6]** modified Tseng's conference key agreement protocol based on bilinear pairing. In 2009, Huang et al. **[10]** proposed a non- interactive protocol based on DL assumption to improve the efficiency of Tseng's protocol. 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.

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 [8], [19] 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 [14] and this approach is similar to the model used in the IEEE 802.11i standard [11] 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. [14] proposed the first algorithm based on
this approach using any (t,n) secret sharing scheme to distribute a group key to a group consisting of (t-1)members. Later, there are some papers [15], [19] 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.

3. Goals of Our Proposed Model

The following are the goals of our proposed authenticated group key transfer model. They are as follows:

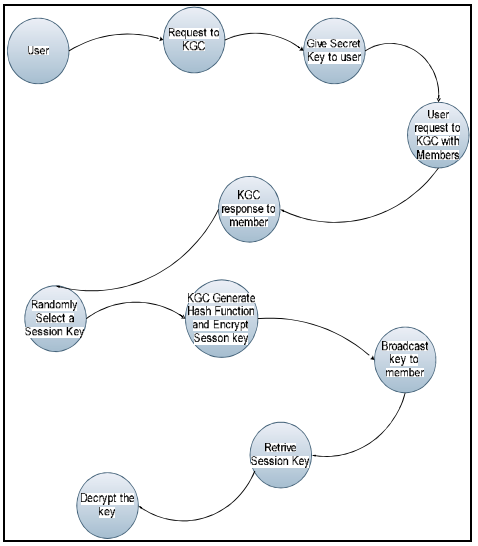
1) Key freshness
2) Key confidentiality; and
3) Key authentication.

Key freshness is to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication.

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

Key authentication is to provide assurance to authorized group members that the group key is distributed by KGC; but not by an attacker.

In our protocol, we only focus on protecting group key information broadcasted from KGC to all group members. The service request and challenge messages from users to KGC are not authenticated. Thus, 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. We need to analyze security threats caused by these attacks. In our security analysis, we will conclude that none of these attacks can successfully attack to authorized group members since attackers can neither obtain the group key nor share a group key with authorized group members. User/message authentication and key confirmation can be easily incorporated into our protocol since each user has shared a secret key with KGC during registration. However, these security features are beyond the scope of our fundamental protocol. We will briefly discuss ways to provide user authentication, message authentication, and key confirmation in security analysis.



4. Authenticated Group Key Transfer 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:

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 an broadcast channel. For example, we assume that a group consists of t members, {U1, U2; . . . ; Ut}, and shared secrets are (xi, yi), for I = 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, R1 E Z_n to KGC.

Step 4. KGC randomly selects a group key, k, and generates an interpolated polynomial

Step 5. For each group member, Ui, knowing the shared secret $(x_i, y_i \oplus R_i)$ and t additional public points, Pi, for I = 1; . . . ; t, on f(x), is able to compute the polynomial f(x) and recover the group key k = f (0). Then, Ui computes h (k, U1, U2….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.

Algorithm Procedure

Step	KGC	Users
1	Group key request $\{A, B, C\}$ ←	Initiator
2	Group key response $\{A, B, C\}$ →	A, B, C
3	R_A ←	A
	R_B ←	B
	R_C ←	C
4	KGC computes $f(x)$ passing through $(0, k)$, $(x_A, y_A \oplus R_A)$, $(x_B, y_B \oplus R_B)$, $(x_C, y_C \oplus R_C)$. KGC also computes P_1, P_2, P_3 on $f(x)$ and $Auth = h(k, A, B, C, R_A, R_B, R_C, P_1, P_2, P_3)$. $Auth, P_1, P_2, P_3$ →	A, B, C
5	Each participating user U_i computes an interpolating polynomial $f(x)$ passing through P_1, P_2, P_3 and $(x_i, y_i \oplus R_i)$. U_i checks whether $Auth = h(k, A, B, C, R_A, R_B, R_C, P_1, P_2, P_3)$.	

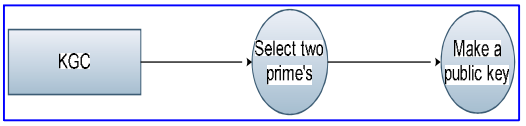
5. Implementation Modules

Implementation is the stage where the theoretical design is automatically converted into practically by dividing this into various modules. We have implemented the current application in Java Programming language with Front End as java Swings, and Back End as SQL Server 2000 data base. Our proposed application is divided into following 4 modules. They are as follows:

1) Initialization of KGC Module
2) User Registration
3) Session Key Generation and Distribution
4) Encryption and Decryption

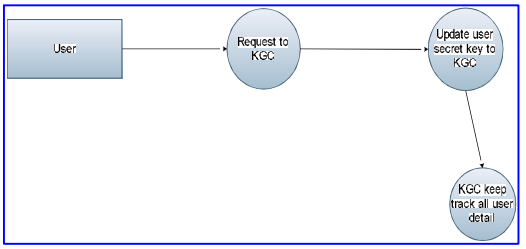
1. Initialization of KGC Module

The KGC randomly chooses two safe primes p and q (i.e., primes such that p1 = (p-1)/2 and q1 = (q-1)/2 are also primes) and compute n = pq. n is made publicly known.



2. User Registration Module

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 (xi yi) with each user Ui, where xi yi € z.



3. 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 is in a broadcast channel. For example, we assume that a group consists of t members,{u1,u2…ut} and shared secrets are (xi yi) for I =1, . . . ,t.

4. Encryption and Decryption

In this module we are encrypting the messages using the session key. After encrypt the message it will be forwarded to selected neighbors. The neighbors get the encrypted message using the session key it will be decrypt the messages.

6. Conclusion

In this paper, we have proposed an efficient group key transfer protocol based on secret sharing. Every user needs to register at a trusted KGC initially and pre share 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.

7. References

[1] H. Harney and C. Muckenhirn, “Group Key Management Protocol (GKMP) Specification,” RFC 2093, July 1997.

[2] D.M. Wallner, E.J. Harder, and R.C. Agee, “Key Management for mMulticast: Issues and Architectures,” RFC 2627, 1999.

[3] C.K. Wong, M.G. Gouda, and S.S. Lam, “Secure Group Communications Using Key Graphs,” IEEE/ACM Trans. Networking, vol. 8,no. 1, pp. 16-30, Feb. 2000.

[4] S.S. Kulkarni and B. Bruhadeshwar, "Adaptive Rekeying for Secure Multicast," IEEE/IEICE Trans. Comm., special issue on comm., vol. E86-B, no. 10, pp. 2948-2956, Oct. 2003.

[5] C. Boyd, "On Key Agreement and Conference Key Agreement," Proc. Second Australasian Conf. Information Security and Privacy (ACISP '97), pp. 294-302, 1997.

[6] J.C. Cheng and C.S. Laih, "Conference Key Agreement Protocol with NonInteractive Fault-Tolerance Over Broadcast Network," Int'l J. Information Security, vol. 8, no. 1, pp. 37-48, 2009.

[7] W. Diffie and M.E. Hellman, "New Directions in Cryptography," IEEE Trans. Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.

[8] A. Fiat and M. Naor, "Broadcast Encryption," Proc. 13th Ann. Int'l Cryptology Conf. Advances in Cryptology (Crypto '93), pp. 480-491, 1994.

[9] H. Harney, C. Muckenhirn, and T. Rivers, "Group Key Management Protocol (GKMP) Architecture," RFC 2094, July 1997.

[10] K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai, and T.S. Chen, "A Conference Key Agreement Protocol with Fault-Tolerant Capability," Computer Standards and Interfaces, vol. 31, pp. 401-405, Jan. 2009.

[11] IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.

[12] I. Ingemarsson, D.T. Tang, and C.K. Wong, "A Conference Key Distribution System," IEEE Trans. Information Theory, vol. IT-28, no. 5, pp. 714-720, Sept.1982.

[13] J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," J. Cryptology, vol. 20, pp. 85-113, 2007.

[14] 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.

[15] C.H. Li and J. Pieprzyk, "Conference Key Agreement from Secret Sharing," Proc. Fourth Australasian Conf. Information Security and Privacy (ACISP '99), pp. 64-76, 1999.

[16] A. Perrig, D. Song, and J.D. Tygar, "Elk, A New Protocol for Efficient Large- Group Key Distribution," Proc. IEEE Symp. Security and Privacy, pp. 247-262, 2001.

[17] M.O. Rabin, "Digitized Signatures and Public-Key Functions As Intractable As Factorization," Technical Report LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[18] 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.

[19] G. Saze, "Generation of Key Predistribution Schemes Using Secret Sharing Schemes," Discrete Applied Math., vol. 128, pp. 239-249, 2003.

[20] A. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, no. 11, pp. 612- 613, 1979.

[21] A.T. Sherman and D.A. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Trans. Software Eng., vol. 29,no. 5, pp. 444-458, May 2003.

[22] D.G. Steer, L. Strawczynski, W. Diffie, and M.J. Wiener, "A Secure Audio Teleconference System," Proc. Eighth Ann. Int'l Cryptology Conf. Advances in Cryptology (Crypto '88), pp. 520-528, 1988.

[23] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," Proc. Third ACM Conf. Computer and Comm. Security (CCS '96), pp. 31-37, 1996.

[24] D.R. Stinson, Cryptography Theory and Practice, second ed., CRC Press, 2002.

8. About the Authors

Appikatla Srihari is currently pursuing his 2 Years M.Tech (Software Engineering) in Department of Computer Science and Engineering at Vignan Institute of Information Technology, Visakhapatnam. His area of interests includes Networks.

Dr.Koduganti Venkata Rao is currently working as Professor and Head of the Department in Department of Computer Science and Engineering at Vignan Institute of Information Technology, Visakhapatnam. His research interests include Security and Cryptography, Parallel Computing & Grid Computing