ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

Group Key Agreement with Restricted Connectivity

MANOJKUMAR.D [1], DR.C.SUMITHRADEVI [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

Abstract — In this paper, we study a group key agreement problem where a user is only aware of his neighbors while the connectivity graph is arbitrary. In our problem, there is no centralized initialization for users. A group key agreement with these features is very suitable for social networks. Under our setting, we construct two efficient protocols with passive security. We obtain lower bounds on the round complexity for this type of protocol, which demonstrates that our constructions are round efficient. Finally, we construct an actively secure protocol from a passively secure one.

Index Terms—Group key agreement, Diffie-Hellman, lower bound, authentication, protocol.

1.INTRODUCTION

Key agreement is a mechanism that allows two or more parties to securely share a secret key (called a session key). Almost all the protocols assume a complete connectivity graph: any two users can communicate directly. In the real world, this is not always true. For instance, in social networks such as Facebook, Skype, Wechat and Google+, a user is only connected with his friends. For a group of users (e.g., the faculty union in a university) who wish to establish a session key, it is not necessary that any two of them are friends. But they might still be connected indirectly through the friend network. Of course, they can still regard those as directly connected by regarding the intermediate users as routers. However, this is quite different from a direct connection. First, indirectly connected users may not have the public information of each other (e.g., public-key certificate). Second, indirectly connected users may not know the existence of each other (e.g., in our faculty union example, one professor in one department may not know another professor in a different department). Third, a message between two indirectly connected users travels a longer time than that between directly connected users.

The study is on the group key agreement with an arbitrary connectivity graph, where each user is only aware of his neighbors and has no information about the existence of other users. Further, he has no information about the network topology. Under this setting, a user does not need to trust a user who is not his neighbor. Thus, if one is initialized using PKI, then he need not trust or remember public-keys of users beyond his neighbors. A group key agreement with a local connectivity is studied where a user is only aware of his neighbors while the connectivity graph is arbitrary. There is no central authority to initialize users. Each of them can be initialized independently using PKI. A group key agreement for this setting is very suitable for applications such as a social network. Under our setting, it constructs two efficient passively secure protocols. It also proves lower bounds on the round complexity which demonstrates that the protocols are round efficient. Finally, the system constructs an actively secure protocol from a passively secure one.

2. RELATED WORKS

Key pre-distribution system (KPS) (a.k.a. non-interactive conference distribution system) [4],can be regarded as a non-interactive group key agreement. In this case, the shared key of a given group is fixed after the setup. If a group is updated, then the group key changes to the shared key of the new group. The drawback of KPS is that the user key size is combinatorially large in the total number of users (if the system is unconditionally secure). Another drawback is that the group key of a given group can not be changed even if it is leaked unexpectedly (e.g., cryptanalysis of ciphertexts bearing this key). The key size problem may be overcome if a computationally secure system is used, while the key leakage problem is not easy. Further, computationally secure KPS is only known for the twoparty case and the three-party case. KPS with a group size greater than 3 is still open [9]. A broadcast encryption is a mechanism that allows a sender to send a group key to a selected set of users. This can be regarded as a group key agreement of one message that is sent by the sender. In a symmetric key based broadcast encryption [10], the sender is a fixed authority. In this case, the user key size is combinatorially lower bounded [10]. In addition, it is secure only against a limited number of users. [6] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). Further, users are initialized by a central authority which is not desired in our setting. Traitor tracing [5], [7], [8] is a special broadcast encryption, where besides the usual broadcast capability, it can trace a pirate user: if a user helps build an illegal decryption device, he will be identified. This primitive inherits the drawbacks of a broadcast encryption. A rekey scheme for a multicast can be regarded as a centralized dynamic broadcast encryption, where the authority always maintains the group asthe set of dynamically changing privileged users, and updates the group key and some user keys whenever there is a membership change. This mechanism has a drawback that the user key is provided by a centralized authority and has to be updated upon a member leave. If a group key agreement adopts this system, then the user key will be updated whenever the group changes. This is not desired. In addition, in a group key agreement setting, it is possible that several groups might simultaneously require to derive a group key. A rekeying scheme can not handle this case. So we can not adopt a rekeying scheme as a group key agreement. In all of KPS, broadcast encryption, traitor tracing and a rekey scheme, a user key is set up by a single central authority and there is a dependency between the keys of different users. The first three mechanisms also have a threshold for the number of corruptions. In our key agreement problem, a centralized setup is not convenient and it is also impossible to determine a corruption threshold. Hence, they are not reasonable candidates for a group key agreement in our setting.

2.1 EXISTING SYSTEM

In social networking there are many applications which provide the data connectivity, communication, file transfer, sharing, uploading and many other operations. But sometimes there are problems in communication between two unknown authorities. Most of the systems does not support to the direct connectivity

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

of unknown authorities' for communication or data transfer. However the one person is neighbor of another person who cannot get access with their neighbors directly. So sometimes it makes problem connectivity. So this can be helped with the group key agreement to make it possible. Key pre-distribution system (KPS) non-interactive conference distribution system can be regarded as a non-interactive group key agreement. In this case,the shared key of a given griup is fixed after the setup.If a group is updated,then the group key changes to the shared key of the new group. The drawback of KPS is that the user key size is combinatorial large in the total number of users(if the system is unconditionally secure).Another drawback is that the group key of agiven group cannotbe changed even if it is leaked unexpectedly(e.g)cryptanalysis of cipher texts bearing this key).The key size problem may be overcome if a computationally secure system is used,while the key leakage problem Is not easy.Further,computationally secure KPS is only known for the two party case and the three-party case KPs with a group size greater than 3 is still open.

2.1.1 Disadvantages

1. The User key size is combinational large in the total number of users (if the system is unconditionally secure).

2. The group key of given group can not be changed even if it is leaked unexpectedly.

2.2. PROPOSED SYSTEM

A key-agreement protocol is a protocol where one user is only aware of his neighbors. Two or more parties can agree on a key in such a way that both influence the outcome. If properly done, this precludes undesired third parties from forcing a key choice on the agreeing parties. Sender generates key and sends it to receiver. The connection made between is actively secure protocol using passively secure protocol. Protocols that are useful in practice also do not reveal to any eavesdropping party what key has been agreed upon. public-key agreement protocol that meets the above criteria was the Diffie–Hellman key exchange, in which two parties jointly exponentiation a generator with random numbers, in such a way that an eavesdropper cannot feasibly determine what the resultant value used to produce a shared key is. Exponential key exchange in and of itself does not specify any prior agreement or subsequent authentication between the participants. It has thus been described as an anonymous key agreement protocol.

2.2.1 Benfits of Proposed System

- A group key agreement with a local connectivity where a user is only aware of his neighbors while the connectivity graph is arbitrary.
- There is no central authority to initialize users. Each of them can be initialized independently using PKI.

A group key agreement for this setting is very suitable for applications such as a social network.

3. METHODOLODY

Let U = {1.....N}; Ng be the universe of users who are connected by an undirected connected graph GU. Assume the set of neighbors for i 2 U is Ui _ U. It is assumed that the user i knows Ui. We will define a key agreement on any undirected connected subgraph G = (V;E) of GU. The set of neighbors of i in G is denoted by Ni(G). The protocol allows users in V to agree on a shared key. Each user i in the protocol can only send messages to his neighbors Ni(G). Since user i has no knowledge about users other than Ui, it must facilitate him to determine Ni(G): Toward this, it can be assumed that there is a basic description of G (denoted by basic(G)) such that with Ui and basic(G), user i can easily determine Ni(G): basic(G) is determined by the protocol initiator and it will appear in the first incoming message of any user (other than the initiator) in G, which for simplicity will not be mentioned again later.

Let U = {1.....N}; Ng be the universe of users connected by an undirected connected graph GU, where user i has a neighbor set Ui. Group key agreement with a local connectivity is a mechanism with the following components.

Setup Upon 1^{λ}, a system parameter sp is generated. For each i ∈ U; a public key PKi and a private key SKi are generated. (sp; PK1; ..... ; PKN) is public while SKi is only known to user i:

Key Agreement. For an undirected connected subgraph G = (V;E) of GU, initiated by some I2 V with input basic(G), users in V interact with their neighbors in G and finally all of them derive a group key sk: The protocol is complete: if users in V follow the protocol, they derive the same sk. As an example, let GU be a connected social network and G be a university faculty union who organizes its members on GU. Each professor has his own friend list Ui in GU. Given the name "faculty union", professor i can determine Ni(G), assuming that he knows that who in his friend list is a professor and who is not. Now if a professor wishes the union to compute a union key. He can send the request "faculty union key" to his union neighbors and interact with them, who then continue the similar interaction with their own union neighbors, and so on. Finally, union members can obtain a group key.

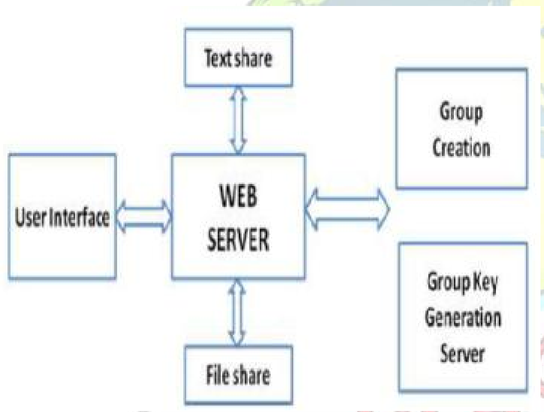

Fig 1. System Architecture

3.1 MODULES

3.1.1 Network Environment Setup

In the first module, we create the network environment setup with nodes, certificate authority. Network environment is set up with nodes connected with all and using socket programming in java.

3.1.2 Key Broadcast

In this module formally define the model of group key agreement-based broadcast encryption. The definition incorporates the up-to-date definitions of group key agreement and public-key broadcast encryption. Since the core of key management is to securely distribute a session key to the intended receivers, it is sufficient to define the system as a session key encapsulation mechanism. Then, the sender can simultaneously encrypt any message under the session key, and only the intended receivers can decrypt. The new paradigm seems to require a trusted third party as its counterpart in traditional broadcast encryption systems. A closer look shows there is a difference. In a traditional broadcast encryption system, the third party has to be fully trusted, that is, the third party knows the secret keys of all group members and can read any transmission to any subgroup of the members. This kind of fully trusted third party is hard to implement in open networks. In contrast, the third party in our key management model is only partially trusted. In other words, the third party only knows and certifies the public key of each member. This kind of partially trusted third party has been implemented and is known as public key infrastructure (PKI) in open networks.

3.3 Group Key Management

The new key management paradigm ostensibly requires a sender to know the keys of the receivers, which may need communications from the receivers to the sender as in traditional group key agreement protocols. However, some subtleties must be pointed out here. In traditional group key agreement protocols, the sender has to simultaneously stay online with the receivers and direct communications from the receivers to the sender are needed. This is difficult for a remote sender. On the contrary, in our key management paradigm, the sender only needs to obtain the receivers' public keys from a third party, and no direct communication from the receivers to the sender is required, which is implementable with exactly the existing PKIs in open networks. Hence, this is feasible for a remote sender. In our scheme, it is almost free of cost for a sender to exclude a group member by deleting the public key of the member from the public key chain or, similarly, to enroll a user as a new member by inserting that user's public key into the proper position of the public key chain of the receivers. After the deletion/addition of certain member, a new logical public-key ring naturally forms. Hence, a trivial way to enable this change is to run the protocol independently with the new key ring.

If the sender would like to include a new member, the sender just needs to retrieve the public key of this user and insert it into the public key chain of the current receiver set. By repeatedly invoking the member addition operation, a sender can merge two receiver sets into a single group. Similarly, by repeatedly invoking the member deletion operation, a sender can partition one receiver set into two groups. Both merging and partitioning can be done efficiently. In this module shows the deletion of member from the receiver group. Then, the sender and the remaining receivers need to apply this change to their subsequent encryption and decryption procedures.

4. CONCLUSION

The system studied a group key agreement problem, where a user is only aware of his neighbors while the connectivity graph is arbitrary. In addition, users are initialized completely independent of each other. A group key agreement in this setting is very

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

suitable for applications such as social networks. It constructed two passively secure protocols with contributiveness and proved lower bounds on a round complexity, demonstrating that these protocols are round efficient. Finally, we constructed an actively secure protocol from a passively secure one. In the proposed work, it did not consider how to update the group key more efficiently than just running the protocol again, when user memberships are changing. It is not clear how to do this. One can either propose algorithms to the current protocols or construct a completely new key agreement with these features. It can be leaved it as an open question.

FUTURE ENHANCEMENT

In the proposed work, it did not consider how to update the group key more efficiently than just running the protocol again, when user memberships are changing. It is not clear how to do this. One can either propose algorithms to the current protocols or construct a completely new key agreement with these features. It can be leaved it as an open question.

REFERNCES

[1] Y. Amir, Y. Kim, C. Nita-Rotaru and G. Tsudik, "On the Performance of Group Key Agreement Protocols", ACM Trans. Inf. Syst. Secur., vol. 7, no. 3, pp. 457-488, Aug. 2004.

[2] D. Augot, R. Bhaskar, V. Issarny and D. Sacchetti, "An Efficient Group Key Agreement Protocol for Ad Hoc Networks", Proc. 6th IEEE Int'l Symp. on a World of Wireless Mobile and Multimedia Networks (WOWMOM 2005), pp. 576-580, 2005.

[3] A. Beimel and B. Chor, "Communication in Key Distribution Schemes", Proc. Advances in Cryptology (CRYPTO'93), vol. 773, pp. 444-455, 1994.

[4] R. Blom, "An Optimal Class of Symmetric Key Generation Systems", Proc. Advances in Cryptology-EUROCRYPT'84, vol. 209, pp. 335-338, 1984.

[5] D. Boneh and M. K. Franklin, "An Efficient Public-key Traitor Tracing Scheme", Proc. Advances in Cryptology (CRYPTO'99), vol. 1666, pp. 338-353, 1999.

[6] Christo Ananth, M.Danya Priyadharshini, "A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks", International Journal of Applied Engineering Research (IJAER),

Volume 10, Special Issue 2, 2015,(1250-1254)

[7] D. Boneh, A. Sahai and B. Waters, “Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys”, Proc. 25th Int’l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT’06), vol. 4004, pp. 573-592, 2006.

[8] D. Boneh and M. Naor, “Traitor Tracing with Constant Size Ciphertext”, Proc. 15th ACM Conf. Computer and Comm. Security, pp. 501-510, 2008.

[9] D. Boneh and A. Silverberg, “Applications of Multilinear Formsto Cryptography”, Contemporary Mathematics, Vol. 324, American Mathematical Society, pp. 71-90, 2003.

[10] C. Blundo, L. A. Mattos and D. R. Stinson, “Generalized Beimel-Chor Schemes for Broadcast Encryption and Interactive Key Distribution”, Theor. Comp. Sci., vol. 200, no. 1-2, pp. 313-334, 1998.

[11] C. Blundo and A. Cresti, “Space Requirements for Broadcast Encryption”, Proc. Advances in Cryptology - EUROCRYPT 1994, vol. 950, pp. 287-298, 1995.

[12] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, “Perfectly Secure Key Distribution for Dynamic Conferences”, Inf. Comput., vol. 146, no. 1, pp. 1-23, 1998.

[13] C. Boyd and J. M. Gonz´alez-Nieto, “Round-Optimal Contributory Conference Key Agreement”, Proc. Public Key Cryptography (PKC’03), vol. 2567, pp. 161-174, 2003.