Key Management Scheme for Efficient Group Communication Networks

Anushiya.K[*], Pradeepa.D[**]

[*] II M.E. C&C, P.B.College of Engineering, Chennai – 602 105
[**] Asst. Prof / ECE, P.B.College of Engineering, Chennai – 602 105

Abstract- In most of the emerging networks the main problem is based on broadcasting a data to a remote cooperative group in efficient and secure manner. To overcome the limited communication from the group to the sender, the unavailability of fully trusted key generation center and the dynamics of the sender we proposing a hybrid of traditional broadcast encryption and group key agreement. In this system, each member maintains a single public/secret key pair. When seeing the public keys of the members, a remote sender can securely broadcast to any subgroup. Following this model, we introduce a scheme that is proven secure in the standard model. In this, even if non-intended members collude, they cannot extract any useful information from the transmitted messages. After the public group encryption key is extracted, both the computation overhead and the communication cost are independent of the group size. It is very strong security against collusion, constant overhead, and also our protocol is a solution to many applications.

Index Terms- Access control, broadcast, cooperative computing, key management.

I. INTRODUCTION

In many newly emerging networks, there is a need to broadcast to remote cooperative groups using encrypted transmission. Examples can be found in access control in remote group communication arising in wireless mesh networks (WMNs), mobile *ad hoc* networks (MANETs), vehicular *ad hoc* networks (VANETs), etc. WMNs have been recently suggested as a promising low-cost approach to provide last-mile high-speed Internet access. A typical WMN is a multihop hierarchical wireless network. The top layer consists of high-speed wired Internet entry points. The second layer is made up of stationary mesh routers serving as a multihop backbone to connect to each other and Internet via long-range high-speed wireless techniques. The bottom layer includes a large number of mobile network users. The end-users access the network either by a direct wireless link or through a chain of other peer users leading to a nearby mesh router; the router further connects to remote users through the wireless backbone and Internet. Security and privacy issues are of utmost concern in pushing the success of WMNs for their wide deployment and for supporting service-oriented applications. For instance, a manager on his way to holiday may want to send a confidential e-mail to some staff of her company via WMNs, so that the intended staff members can read the e-mail with their mobile devices.

A MANET is a system made up of wireless mobile nodes. These nodes have wireless communication and networking characteristics. MANETs have been proposed to serve as an effective networking system facilitating data exchange between mobile devices even without fixed infrastructures. In MANETs, it is important to support group-oriented applications, such as audio/video conference and one-to-many data dissemination in battlefield or disaster rescue scenarios. VANETs are designed with the primary goal of improving traffic safety and the secondary goal of providing value-added services to vehicles.

II. EXISTING AND PROPOSED WORK

In this paper the proposed scheme is new key management paradigm, allow secure and efficient transmissions to remote cooperative groups by effectively exploiting the mitigating features and circumventing the constraints discussed above. The new approach is a hybrid of group key agreement and public-key broadcast encryption. In our approach, each group member has a public/secret key pair. By knowing the public keys of the members (e.g., by retrieving them from a public key infrastructure that is widely available in existing network security solutions), a remote sender can securely broadcast a secret session key to any intended subgroup chosen in an *ad hoc* way and simultaneously, any message can be encrypted to the intended receivers with the session key. Only the selected group members can jointly decrypt the secret session key and hence the encrypted message. In this way, the dependence on a fully trusted key server is eliminated. Also, the dynamics of the sender and the group members are coped with because the interaction between the sender and the receivers before the transmission of messages is avoided and the communication from the group members to the remote sender is minimized. For secure transmission to remote cooperative groups, a core is to establish a one-to-many channel securely and efficiently under certain constraints. We observe that the existing key management approaches do not provide effective solutions to this problem. On one hand, group key agreement provides an efficient solution to secure intra group communication, but for a remote sender, it requires the sender to simultaneously stay online with the group members for multiple rounds of interactions to negotiate a common secret session key before transmitting any secret contents. This is impractical for a remote sender who may be in a different time zone. This situation is further deteriorated if the sender is mobile or otherwise dynamic. On the other hand, broadcast encryption enables external senders to broadcast to non cooperative members of a preset group without requiring the

sender to interact with the receivers before transmitting secret contents, but it relies on a centralized key server to generate and distribute secret keys for each group member.

III. PROBLEM STATEMENT

We consider a group composed of N users, indicated by $\{U_1,..,U_N\}$. A sender would like to transmit secret messages to a receiver subset S of the N users, where the size of S is n<=N. The problem is how to enable the sender to efficiently and securely finish the transmission with the following constraints.

- It is hard to deploy a key generation authority fully trusted by all users and potential senders in open network settings.
- The communication from the receivers to the sender is limited, e.g., in the battlefield communication setting.
- N might be very large and up to millions, for instance, in vehicular ad hoc networks.
- Both the sender and the receiver sets are dynamic due to *ad hoc* communication.

According to the application scenarios, there are also some mitigating features that may be exploited for solving the problem.

- n is usually a small or medium value, e.g., less than 256.
- The receivers are cooperative and communicated via efficient local (broadcast) channels.
- A partially trusted authority, e.g., a public key infrastructure, is available to authenticate the receivers (and the senders).

IV. SYSTEM ARCHITECTURE

The system architecture is illustrated in figure. The potential receivers are connected together with efficient local connections. Via communication infrastructures, they can also connect to LAN networks. Each receiver has a public/secret key pair. The public key is certified by a certificate authority, but the secret key is kept only by the receiver. A remote sender can retrieve the receiver's public key from the certificate authority and validate the authenticity of the public key by checking its certificate, which implies that no direct communication from the receivers to the sender is necessary. Then, the sender can send secret messages to any chosen subset of the receivers. The System contains a sender and a number of receivers to receive the message. The sender has the authority control to enable the set of receivers those who are capable to receive the message. When the sender needs to send a message to a group of receivers he checks the public key of all receivers with the certificate authority.

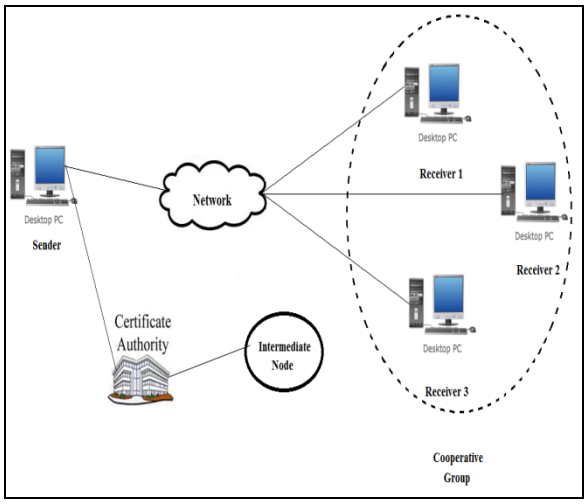

Fig. 1. System Architecture

The certificate authority provides the exact public key of all set of receivers to the sender. The sender proceeds with further action only after the verification of the public key of the receivers. The purpose of our proposed system is to pass message with security to multiple receivers .This implements secured system using a secrete key for individual receivers such that the receiver can decrypt the message only with the secret key. By chance if an unauthorized person receives the message it remains in the encrypted format.

Main MODULE

The following list of modules are used, they are

- Access Control
- Certificate Authority
- Information Security

Access Control

The System contains a sender and a number of receivers to receive the message. The sender has the authority control to enable the set of receivers those who are capable to receive the message. A more narrow definition of access control is to only cover access approval, whereby the system makes a decision to grant or reject an access request from an already authenticated subject, based on what the subject is authorized to access.

Authentication methods include passwords, biometric scans, physical keys, electronic keys and devices, hidden paths, social barriers, and monitoring by humans and automated systems. Access control systems provide the essential services of authorization, and authentication, access approval.

Certificate Authority

When the sender needs to send a message to a group of receivers he checks the public key of all receivers with the certificate authority. The certificate authority provides the exact public key of all set of receivers to the sender. The sender proceeds with further action only after the verification of the public key of the receivers. A Certificate Authority issues digital certificates that contain a public key and the identity of the owner. The matching private key is not made available publicly, but kept secret by the end user who generated the key pair. The certificate is also a confirmation or validation by the Certificate

Authority that the public key contained in the certificate belongs to the person, organization, server or other entity noted in the certificate. A Certificate Authority's obligation in such schemes is to verify an applicant's credentials, so that users and relying parties can trust the information in the Certificate Authority's certificates. Certificate Authority's use a variety of standards and tests to do so. In essence, the certificate authority is responsible for saying yes, this person is who they say they are, and we, the Certificate Authority, certify that. If the user trusts the Certificate Authority and can verify the Certificate Authority's signature, then he can also assume that a certain public key does indeed belong to whoever is identified in the certificate.

Information Security

This implements secured system using a secrete key for individual receivers such that the receiver can decrypt the message only with the secret key. By chance if an unauthorized person receives the message it remains in the encrypted format. Two major aspects of information security are:

IT security: Sometimes referred to as computer security, Information Technology Security is information security applied to technology. It is worthwhile to note that a computer does not necessarily mean a home desktop. A computer is any device with a processor and some memory (even a calculator). IT security specialists are almost always found in any major enterprise/establishment due to the nature and value of the data within larger businesses. They are responsible for keeping all of the technology within the company secure from malicious cyber attacks that often attempt to breach into critical private information or gain control of the internal systems.

Information assurance: The act of ensuring that data is not lost when critical issues arise. These issues include but are not limited to: natural disasters, computer/server malfunction, physical theft, or any other instance where data has the potential of being lost. Since most information is stored on computers in our modern era, information assurance is typically dealt with by IT security specialists. One of the most common methods of providing information assurance is to have an off-site backup of the data in case one of the mentioned issues arise.

V. KEY MANAGEMENT

The Key management two type of the management each receiver has a public/secret key pair. The public key is certified by a certificate authority, but the secret key is kept only by the receiver. A remote sender can retrieve the receiver's public key from the certificate authority and validate the authenticity of the public key by checking its certificate, which implies that no direct communication from the receivers to the sender is necessary. Then, the sender can send secret messages to any chosen subset of the receivers. 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. 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. A sender does not need to frequently contact the third party or keep a large number of keys since a sender usually communicates to a relatively fixed group in practice. For instance, a department manager usually communicates with her subordinates, superiors, and other department managers, but rarely needs to send secret messages to all staff members.

VI. SYSTEM IMPLEMENTATION

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. Specifically, our key management system consists of the following (probabilistic) polynomial-time algorithms.

Key Generation: This key generation algorithm is run by each user U_i to generate her public/private key pair. A user takes as input the system parameters n, N and her index i, and outputs (pk_i,sk_i) as her public/secret key pair. Denote by $\{(pk_i,sk_i),(u_1\ldots,u_N)\}$. Actually, n,N are polynomials. We assume that each user's public key is certified by a publicly accessible certificate authority so that anyone can retrieve the public keys and verify their authenticity. This is plausible as public key infrastructures have been a standard component in many systems supporting security services. The key generation and the registration to the certificate authority can be done offline before the online message transmission by the sender.

Encryption: It is run by any sender who may or may not be in $(u_1\ldots,u_N)$, provided that the sender knows the public keys of the potential receivers. It takes as input a recipient set S and the public key pk_i for U_i. If S=n, it outputs a pair (Hdr,k), where Hdr is called the header and k is the message encryption key. (S,Hdr) is sent to the receivers. This algorithm incorporates the functionality of the encryption procedure in traditional broadcast encryption systems.

Decryption: This algorithm is jointly run by the intended receivers to extract the secret session key hidden in the header. Each receiver U_j privately inputs her secret key sk_j. The common inputs are the header Hdr and the public keys of receivers in the recipient set S. If S=n, each receiver in S outputs the same session key k. This procedure incorporates a traditional group key agreement protocol. It exploits the cooperation of the receivers with efficient local connections.

We next justify the assumptions on trusted authorities and limited communication from the receivers to the sender in our key management paradigm. At a first look, 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. Second, 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. From the definition, only the sender and the intended receivers are involved in the and procedures. Hence, the complexity of the system does not depend on the size of the full group, but on the size of the receiver subset. The same analysis applies to the dynamics of the sender and the receivers. This implies that our approach is particularly efficient in the case when the full group is very large but the actual receiver set is small. Indeed, our protocol enjoys almost constant complexity when coping with the change of the sender or the receivers. This is especially attractive for mobile networks

VII. SECURITY ANALYSIS

When focusing on the confidentiality of the session key transmitted by the sender, we implicitly assume that the public keys of users are authentic, that is, we assume that they have been previously authenticated. We start by defining the correctness of our system as the property that any user in the receiver set can decrypt a valid header. A formal definition follows.

Correctness

Key encapsulation mechanism (KEM) sends a (short) secret session key to the intended receivers and, simultaneously, (long) messages can be encrypted under the session key using a secure symmetric encryption algorithm. Formally, secrecy is defined by means of the following game between an attacker and a challenger. Both and are given as input, where are polynomials in the security parameter.

Setup: The challenger runs to obtain the users' public keys. The challenger gives the public keys and public system parameters to the attacker.

Corruption: Attacker adaptively issues private key queries.

Challenge: At some point, the attacker specifies a challenge set, satisfying the private key of any user queried in the corruption step.

Observation: After receiving the challenge header, the attacker can access the public transcripts from users in during the decryption interactions.

Guess: Attacker outputs a guess bit for wins.

Secrecy

We say that a group key agreement-based broadcast encryption scheme is collusion-resistant against adaptive attacks if for any polynomial-time attacker.

VIII. OTHER ISSUES

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.

Member Organization

Many key management schemes organize the users in a tree-based structure. However, for our scheme, it is preferable to organize them in a chain and then use the sender to close the chain to form a logical ring. The chain can be formed by ordering the users lexicographically by the least important bits of their unique public keys, and then a ring is formed by closing the chain with the sender as illustrated in Fig. 2, where the public keys of the receivers and the temporary public key of the sender appear as the corresponding nodes in the ring, respectively. Compared to the tree-based structure, the above structure allows better performance for receiver and sender changes. Without loss of generality, we generate sorted chain. In this way, if the sender changes, only receivers and need to communicate with other receivers during the decryption procedure if the receiver set does not change (functionally, this is equivalent to updating the group decryption key of the receivers, so that the previous sender cannot read the message transmitted by the current sender. This is very desirable if the sender may change frequently while the local cooperative group is relatively static.

B. Member Deletion/Addition and Group Partition/Merging

In existing group key agreement-based key management protocols, to exclude a group member or enroll a new member, multiple rounds of communication among the members are required *before* the sender can securely broadcast to the new receiver set. 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.

a. *Member Deletion:* Fig. 3 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.

Encryption: The sender runs this algorithm as follows.

1) Randomly select sender and compute key.

In this step, the sender indexed by reinserts herself into the ring and connects to receivers. Hence, the operation is the same as that of the basic protocol, but the sender has to choose new random values.

2) Compute the new public group encryption key. Since member is deleted, receiver then plays the role of the deleted receiver and connects to receiver.

The following three steps are to compute the session key and the header.

3) Compute S

4) Compute the new secret session key.
5) Broadcast to the receivers the new header.

Decryption: The receivers run this algorithm as follows.

1) According to Step 1 of the procedure of the basic protocol, it is easy to see that only receivers and need to respond to the change in this step.

2) Each receiver indexed, and computes the new group decryption key.

In the above, due to deletion of receiver, remaining receiver in the receiver set.

3) Each receiver extracts the new session key.

b. Member Addition: 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. Fig. 4 shows the addition of member to the receiver group.

Then, the sender and receivers in the new receiver set need to apply this change to their subsequent encryption and decryption procedures.

Encryption: The sender runs this algorithm as follows.
1) Randomly select sender.
2) Compute the new public group encryption key
3) Compute S
4) Compute the new secret session key
5) Broadcast the new header.

Decryption: The intended receivers run this algorithm as follows.
1) Only receivers need to respond to the change in this step.
2) Each receiver indexed, and computes the new secret decryption key.
3) Each receiver extracts the new session key.

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.

C. Rekeying

The above refers to the change of members. Even if the receiver group does not change, various scenarios may require key update. This is a complex issue in most key management schemes.

Session Key Update: This first level is to update the session key. This key is used to encrypt digital contents to the receivers, and it expires after each session.

Group Decryption Key Update: The second level is to update the secret decryption key used by the receivers to compute the session key.

Long-Term Secret Key Update: The third level is to update the secret key of user. This is needed if the user's public key expires or is compromised.

IX. CONCLUSION

The new paradigm seems to require a trusted third party as its counterpart in traditional broadcast encryption systems. 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 partially trusted third party has been implemented and is known as Public Key Infrastructure (PKI) in open networks. 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. 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.

REFERENCES

[1] Y. Zhang and Y. Fang, "ARSA: An attack-resilient security architecture for multi-hop wireless mesh networks," IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1916–1928, Oct. 2006.

[2] K. Ren, S. Yu, W. Lou, and Y. Zhang, "PEACE: A novel privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks," IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 2, pp. 203–215, Feb. 2010.

[3] B. Rong, H.-H. Chen, Y. Qian, K. Lu, R. Q. Hu, and S. Guizani, "A pyramidal security model for large-scale group-oriented computing in mobile ad hoc networks: The key management study," IEEE Trans. Veh. Technol., vol. 58, no. 1, pp. 398–408, Jan. 2009.

[4] Y.-M. Huang, C.-H. Yeh, T.-I. Wang, and H.-C. Chao, "Constructing secure group communication over wireless ad hoc networks based on a virtual subnet model," IEEE Wireless Commun., vol. 14, no. 5, pp. 71–75, Oct. 2007.

[5] Q.Wu, J. Domingo-Ferrer, and U. González-Nicolás, "Balanced trustworthiness, safety and privacy in vehicle-to-vehicle communications," IEEE Trans. Veh. Technol., vol. 59, no. 2, pp. 559–573, Feb. 2010.

AUTHORS

First Author – Anushiya.K, II M.E. C&C, P.B.College of Engineering, Chennai – 602 105,
Email_id:anushiya_k@yahoo.com.
Second Author – Pradeepa.D, Asst. Prof / ECE, P.B.College of Engineering, Chennai – 602 105,
Email_id:pradeepa.do5@gmail.com.