IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 05, Issue 06 (June. 2015), ||V1|| PP 55-59

Efficient Secured Model For Communication In Dynamic Multicast Groups

Prof. Shilpa Harnale
Asst. Professor, Dept. of CSE BKIT, Bhalki

Abstract: Many network applications are based on a group communications model where one party sends messages to a large number of authorized recipients and/or receives messages from multiple senders. In this paper we propose a series of novel approaches for achieving scalable security in IP multicast, providing privacy and authentication on a group-wide basis. They can be employed to efficiently secure multi-party applications where members of highly dynamic groups of arbitrary size may participate. The main idea is to have group members actively participate to the security of the multicast group, therefore reducing the communication and computation load on the source. Since the group security is distributed among the group members, A communication model is proposed for dynamic multicast groups which is secure and authenticated. In this scheme a group is established and the text messages are transmitted between the users of the group.

I. INTRODUCTION

IP multicast is used to distribute data to a group of receivers efficiently. A Datagram addressed to the multicast group, identified by a Class D IP address, will be delivered to all group members. Efficiency can be achieved because datagram's need to be transmitted once and they traverse any link between two nodes only once, saving the cost of sender as well as network Bandwidth.

The main problem in secure multicast is access control for making sure that only legitimate members of multicast group have access to the group Communication. The security algorithms applicable to unicasting (one to one) environment cannot be applied in multicast groups. The commonly used technique in secure multicast is to maintain a group key that is known to all users in the multicast group, but is unknown to others outside the group. Each time a member either joins or leaves the group, the group key has to be refreshed. The members of the group should be able to compute the new group key efficiently, guaranteeing forward and backward secrecy simultaneously. In a dynamic secure multicast, the group key need to be refreshed very frequently. Once a communication group is formed, the members in a group can send/receive messages from other members of same/different groups.

II. RELATED WORK

Existing protocols for secure multicasting are limited to distribute session keys in static and/or small groups.

For dealing with the group key distribution in a large group with frequent membership changes, some good explorations have been done:

Spanning Tree [BD96] proposes the distribution of the key along a spanning tree generated between the members. It relies on trust in all members to forward the data without modification and does not handle group membership changes securely and efficiently.

Cliques The approach proposed in [STW97] is to improve the capability of a system to distribute session keys in dynamic groups, but the solution does not scale well to large groups, since the group manager has to perform O(n) exponentiations for each group membership change and messages get prohibitively large.

Iolus In Iolus[Mit97], a large group is decomposed into a number of subgroups, thus reducing the number of members affected by a key change due to membership changes. It relies on "relay nodes" performing admission control and packet rekeying. This not only requires full trust into these relays, but also increases the transmission delay, and does not handle relay failures gracefully.

Multicast Trees Very recently we came across two schemes for multicast key distribution that are remarkably similar to our own tree-based approach. One is by D. Wallner, E. Harder, and R. Agee, from the National

Security Agency, currently only available as an expired Internet draft. The other scheme, by C. Wong, M. Gouda, and S. Lam, from the University of Texas, is scheduled to appear in SIGCOMM'98.

III. SECURE MULTICASTING

In the solutions presented here, changes to the group's membership are possible with minimal involvement of dedicated nodes and group members, limiting number and size of messages and computing resources needed. The approaches cope with several properties inherent to multicast and broadcast environments: An unreliable (and in the case of IP also unordered) transmission channel and the transmissions may be one-way, with no or only a minimal return channel, to reflect the nature of broadcast environments likely users of secure multicasting. While third party entities such as routers or intermediate systems are entrusted with forwarding secured data, they are not allowed to gain access to actual keying material or plain-text payload. As seen earlier, it is important to have a system which even with large groups and frequent joins or leaves neither is susceptible to implosion nor enables users to understand what was transmitted at times they were not part of the group, either before they joined or after they left or were expulsed. Additionally, any third party recording ongoing transmission and later capturing the secrets held by a participant must not be able to understand its recordings. This is known as "perfect forward secrecy" [Dif90]. To completely achieve this, the unicast connections also need to be set up using ephemeral secrets.

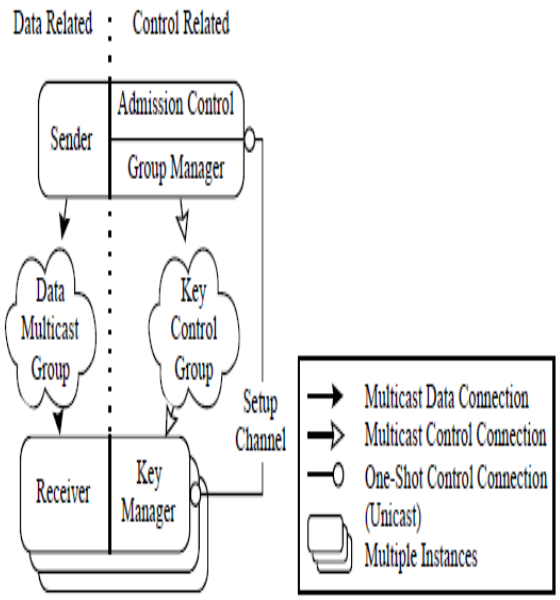


Figure 1: Secure multicasting components in a single sender, multiple recipients' scenario

The basic architecture as shown in the figure the simplest scenario, forming the basis of the descriptions: A single sender and any number of participants (multiple sender and other scenarios will be explained below). Fundamental and common functions are explained here, while individual extensions and modifications will be pursued later. Generally, the components can be separated into two groups: (1) A group of data related components, covering components very similar to those of current insecure multicast or broadcast communication architecture. It consists of the sender, recipients, and one or more Data Multicast Groups. (2) A group of control (or key management) related components, which includes all components involved in the key agreement and key exchange process.

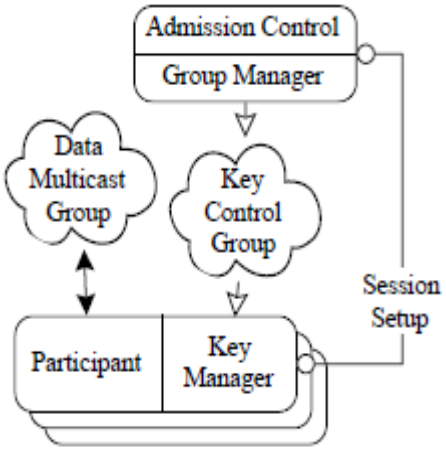


Figure 2: Group collaboration scenario

Often, there is more than one sender, and senders and receivers cannot be distinguished. Also, any receiver is free to send data encrypted or authenticated using the current TEK, and in a group collaboration environment every member of the group holds both roles at the same time, resulting in a situation as shown in Figure 2.

This is a transformation of Figure 1 where sender and recipient were integrated, and the Group Manager has been isolated. All of the schemes also work in that scenario, and the distributed key management scheme even is very well suited for it. If senders and receivers are treated equally, they will be referred to using the term *participant*.

Should a unique, unmistakable identification of the sender be required, as opposed to the identification as an admitted group member, it is necessary for the sender to asymmetrically authenticate each data packet. For many applications, immediate recognition of outsiders injecting traffic is crucial, but it is acceptable to detect sender impersonation by already admitted group members within a certain pre-defined time limit after the fact has occurred. For these applications, it is possible to have the messages authenticated symmetrically (using a MAC) and amortize the costly asymmetric operation over several packets. To achieve this, the sender retains MAC values of all packets sent. In regular time

intervals, it distributes the collected list of MAC values together with a single asymmetric signature over these MACs to the recipients. Thus, the authenticity of all the data packets sent out can be verified by the recipients with a single asymmetric operation, even if they did not get all of the original packets1. This procedure also can be used by the group manager to uniquely authenticate the source of keying material to the group members.

IV. PROPOSED ARCHITECTURE

A. Basic Operations on the Group

To transmit the Traffic Encryption Key (TEK) secretly, a number of Key Encryption Keys (KEKs) are used to encrypt the control traffic containing the TEK. To distinguish the keys, each key consists of a reference tuple containing a unique ID, a version, a revision, and the keying material proper. The key to be used to decrypt a message (or part of it) is always referred to by an (ID, version, revision) tuple. The usage of independent version and revision fields allows *zero-message joining* and is explained below in the leave and joins descriptions, respectively. The abovementioned components and keys will be involved in different activities:

Group Creation The Group Manager is configured with group and access control information. Additionally, the group parameters are published using a directory service.

Single Join The new participant's Key Manager sends its request to the Group Manager, which checks whether this participant is allowed to join. If yes, the GroupManager assigns a unique ID to him, and selects a series of KEKs which will be transmitted to the newcomer. The selection of KEKs will be discussed separately for each key management scheme. The Group Manager now increases the *revision* of all keys (TEK and KEKs) to be

transmitted to the participant by passing the keying material through a one-way function (e.g. a cryptographically secure hash), then sends the keys out to the new participant. It also informs the sender(s) to update their revision and TEK. The other participants will notice the revision change from the key reference tuple in ordinary data packets, and also pass their TEK through the one-way function. Since the function is not reversible, the newcomer has no way to determine the key that was used beforehand.

Single Leave There are three ways to leave a group, namely "Silent Leave", "Voluntary Leave" and "Forced Leave". Only the third kind is of interest here as the first two do not require any action from the group manager. If the Admission Control feels a need to forcibly exclude a participant, a leave message is to be sent out. Also, participants may ask the Admission Control to exclude a member. It is up to the admission policy how to deal with such requests. To exclude a member, all keys known to it need to be replaced with entirely new keying material. To make all remaining participants aware of this change, the key's *version* number is increased. The Group Manager sends out a message with new keying material which can be decrypted by all the remaining participants' Key Managers, but not the member which just left.

Multiple Join, Multiple Leave, Group Merge, Group Split

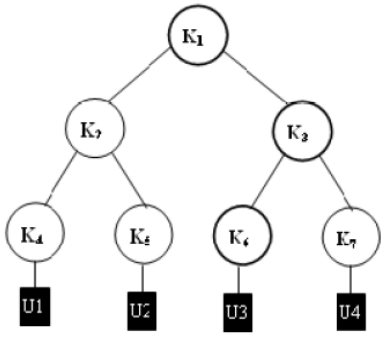
These functions have a number of dependencies on the chosen scheme, and enhance usability of the presented architectures. Due to space constraints, see [CWSP98] for a description.

Group Destruction The Group Manager notifies all remaining participants of the destruction, closes all network connections, destroys all keying material and frees all memory. As soon as all parties have thrown away their keying material, perfect forward secrecy covering all traffic against third party opponents is guaranteed.

B. Centralized, Tree-Based Key management

Tightest control over the individual participants can be achieved by this centralized approach, which is thus suitable for applications with high security demands. It is very easy to implement and maintain, and poses very little load on the network and the receivers. All keying material is managed centrally by the Group Manager, where all joining participants have to register. All keying material is managed centrally by the Group Manager, where

all joining participants have to register. To store the keying material, any tree of arbitrary degree2 can be used. The participants are represented by leaves therein. For simplicity of the explanation assume that the tree is a fully balanced, complete binary tree.



C. Algorithm

The basic idea of our scheme is the usage of an identity tree, where each ode in the tree has an identity. A node in the identity tree is also associated with a key generation key (KGK) which is used for generating a parent key. The root nodes

KGK is used as the group key. Each SGC (Sub Group Controller) constructs an identity tree and gives it to the KGCs (Key Generation Centre). Given a nodes Ni user id, KGC generates the keys for the nodes by ECC algorithm. The generated key is then encrypted using KGK of the user through DES algorithm and encrypted key is transmitted to the user. The user decrypts the key using its KGK, which provides security to the key and saves it in its key ring. Whenever a new user joins the group a new node is added to the group tree with user

name as its identity and GC/SGC requests KGC to refresh the group key and also to generate KGK for the new user. Whenever a user requests to leave the group the GC/SGC removes the node corresponding to user from group tree and KGC is request to
re-key.
Whenever a user wants to send a message to group it generates the message as a text file, it is encrypted using the RSA algorithm and encrypted text file is sent to the SGC/GCI
of the subgroup to which it belongs through TCP/IP connection. SGC/GCI of sub tree to which sender belongs establishes a TCP/IP connection with SGC/GCI of sub tree to which receiver belongs and sends the text containing the encrypted message through socket API. The SGC/CGI of receiver establishes a TCP/IP connection with receiver and transmits the text file to it.
The encrypted text file is decrypted and is added to the mailbox of user. The user can sign in and view its messages. For a user
to sign-in a secret code is generated by GC and provided to the user when it joins the group.

V. CONCLUSION

The mechanism is secured by using RSA algorithm for message encryption or decryption, authenticated as each message contains the identity of the user. The KGC supports large and dynamic multicast groups. Maintaining the users in form of a tree makes this scheme scalable and efficient.

REFERENCES

[1] Liming Wang and Chuan –Kun Wu, " Efficient Key Agreement for Dynamic and Large Multicast Groups", International Journal of Network Security, Institute of Software, Chinese Academy of Sciences, Vol.3, No.1, PP.8–17, July 2006
[2] P.S.L.M. Barreto, H Y Kim and Scott "Efficient algorithms for pairingbased cryptosystems" in CRYPTO 2002, LNCS 2442.
[3] D Bonch and M Franklin, "Identity-based encryption fron weil pairing", in CRYPTO 2001, LNCS 2139.
[4] R Canetti, J Garay, G Itkis, K Micciancio, M Naor and B Pinkas, "Multicast security: a taxanomy and some efficient constuctions", in INFOCOM 1999.
[5] R Canetti, S Halevi and J Katz, "A forward secure public key encryption scheme", in Eurocrypt 2003, LNCS 2656.
[6] I Chang, R Engel, D Pendarakis and D Saha, "Key management for secure Internet multicast" using Boolean function minimization techniques", in INFOCOM 1999.