IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4749

Multicast Security: Issues and New Schemes for Key Management

K.D. Edoh
Department of Computer Science
Montclair State University
Upper Montclair, NJ 07043
Tel: 973 655-5398
Fax: 973 655-4164
edoh@pegasus.montclair.edu

Hussein Abdel-Wahab
Department of Computer Science
Old Dominion University
Norfolk, VA 23529
Tel: 757 683-4512
Fax: 757 683-4900
wahab@cs.odu.edu

ABSTRACT

Security is one of the major concerns for using multicast communications in many Internet applications. This paper identifies and discusses various issues related to secure multicasting. It reviews some requirements for creating secure multicast sessions and gives an overview of existing secure multicast schemes. The main security problem discussed is key management. Taxonomy of various schemes that provide solutions to this problem is given and new improved key distribution schemes are provided. We expand the traditional two basic schemes, the single-group and single-tree, into three new schemes: group-of-trees, tree-of-groups and tree-of-trees. The performance of all these schemes is evaluated as a function of the multicast group size.

1. INTRODUCTION

Multicast is one of the most efficient ways to distribute rich media to multiple users simultaneously over the Internet. It is rapidly becoming an important mode of communication in group-oriented services. Multicasting has many applications, for instance in news groups, chat rooms, teleconferencing, distance education, distributed databases. The group size varies from a small chat group to a very large radio and television broadcast group. In some cases one needs secure multicasting for safe communication within the multicast group. Examples pay-per-view, replicated databases, and chat session among military personnel.

The attacks on multicast security can be considered as part of a general network security attacks. These are usually classified into two useful categories, passive and active attacks. The passive attacks involve eavesdropping or monitoring transmission while the active attacks involve the modification of multicast stream of data or the creation of false stream of data. The requirements in designing secure multicast sessions are to prevent these attacks. They include providing confidentiality, authentication, data integrity, service protection against denial of service attacks, key distribution, specialist requirements, access control, and non-repudiation services.

The most important tool used in multicast security is cryptography [1]. It provides confidentiality, the assurance that multicast data is private to only the intended multicast group and authentication through *digital signatures and public-key certificates*. Cryptography also provides data integrity, that is, data is not altered during transmission using message *integrity codes*. To create a secure multicast session therefore, depends on how best one protects the secret key material in these schemes while preserving multicast efficiency for large groups.

The efficient use of keys in a group context is known as group key management. In [2], a simple xor-based scheme is used to distribute the group key but in general more work is needed to provide a scalable and efficient key distribution schemes for secure multicast communications.

The two most common types of encryption are conventional (symmetric or single key) and the public-key (asymmetric or two-key) encryption. The most popular symmetric encryption algorithms are the data encryption standard (DES) and the triple data encryption algorithm. The public-key encryption algorithms provide message authentication and integrity in addition to message confidentiality. The most popular public-key encryption algorithms are the RSA and Diffie-Hellman algorithms. The other two public-key algorithms are digital signature standard (DSS) and the elliptic curve algorithms.

Next section describes the basic multicasting concepts and services. In section three we give an overview of the traditional group key distribution schemes and then describe our new key distribution schemes. Section four is our conclusion.

2. MULTICAST SERVICES

IP multicast is defined in [4] as a transmission of an IP datagram to a host group. Every multicast group has a group address for example, in IPv4 a Class D IP address. The hosts in the group can receive and transmit IP datagrams to and from the group address respectively. Multicasting does not require the sender to be a member of the recipient group. The groups have technically no group owner. The initiator of the group may advertise the initial setup of the group through Session Announcement Protocol [5] or Session Initiation Protocol [6] to potential participants. These initial announcements include the session security association [7].

A host can join or leave a multicast group by using the Internet Group Management Protocol (IGMP). This protocol enables the host to notify its local router to forward all IP datagrams designated to the multicast address to it. A message sent to a group address is replicated at the routers and forwarded to group members on the network. A distribution tree is created with the routers each of whom maintains the state information about all the interfaces of hosts receiving the multicast packets. The only transport protocol that supports multicast is the User Datagram Protocol (UDP). It is an unreliable datagram service and currently there are several experimental protocols built on top of UDP to provide a more reliable end-to-end transport protocol for multicast sessions. Among the experimental protocols are Scalable Reliable

Multicast (SRM)[8] and Reliable Multicast Transport Protocol (RMTP) [9].

Factors influencing the provision of IP multicast security include the group dynamics and size, multicast application type, security policies, and trust with entities that manage cryptographic keys. The multicast application types are generally categorized as one-to-many, many-to-many or many-to-one. In one-to-many applications a source sends message to several receivers as in multimedia broadcasting of stocks quotes, pay-per-view and in push technology. Security applications that use a centralized security policy tend to favor this category of multicast applications. The lifetime of this group is usually long and the group membership is dynamic with its size varying from several thousands to a million. The recipient's machines usually have limited resources. The authenticity of the transmitted data is very important.

In many-to-one applications several sources send data to one receiver. These applications include resource discovery, data collection and auction. They are favored by security policies that are based on reversed centralized security architectures.

In many-to-many applications like video conferencing and distributed games, several sources send data to several recipients. These applications are favored by distributed security architectures. The group size is usually small from several dozens to a hundred and is not very dynamic. Each member has similar size of computational resources. Authenticity of data is very important.

IP multicast routing protocols [10, 11, 12] are used to route multicast packets. They use the *time-to-live* field in the multicast packet to determine how far to forward the packet. An experimental multicast network with IP multicast test bed is the MBONE. It is a virtual network with IP tunnels between multicast routers. IP datagrams are encapsulated into a second IP datagram and sent to appropriate multicast routers as unicast datagram. MBONE uses an improved version of the Distance Vector Multicast Routing Protocol (DVMRP)[10] for routing its datagrams. A second multicast routing protocol is the Core Based Tree (CBT)[11] routing protocol. This protocol offers a scalable solution to some key management schemes. Other multicast routing protocols include the Multicast extension to OSPF (MOSPF) defined in RFC 1584 and the Protocol Independent Multicast (PIM) defined in RFC 2117.

Multicast applications can make use of IP Security (IPSec) tools to provide security to multicast datagrams. IPSec [7] is an Internet security layer architecture that makes available authentication, confidentiality and key management mechanisms to applications that need it. It is independent of application algorithms.

3. KEY DISTRIBUTION

In order to encrypt the data sent out during a multicast session, every member of the receiver group must have a group key to decrypt the data. The group key may be generated through all the members of the group (called key agreement). It can also be generated by a collection of some group members, or by a trusted Key Distribution Center (KDC). The group key is modified (re-key) if a new member joins the group in order to prevent him from assessing previous session messages. There is also a re-key if a member leaves the group so as to prevent the member from assessing further session messages (member revocation).

In the key agreement schemes the re-key latency (delay) is very large, and not suitable for highly dynamic multicast groups with frequent re-key compared to the KDC models. The key agreement protocols are suitable for many-to-many type of application. However, they are not suitable for applications of the type one-to-many or many-to-one and applications with a heterogeneous environment of varying bandwidth and computation power.

If a symmetric encryption scheme is used to encrypt the data, then all parties in the multicast group should know the shared key, including those intending to send message to the multicast group. If an asymmetric encryption is used instead, then the public keys of all the receiving parties should be distributed to all the sending parties. On the average the public key algorithms are much slower than the symmetric key algorithms.

Two main areas of concern with respect to key management are the initialization of the multicast group with a common net key and re-keying the multicast group.

Classification of key distribution schemes

The two major classifications of key distribution architectures are the single-tree scheme and the single-group scheme described below. We introduced three more efficient schemes which are a combination of the single-tree scheme and the single-group scheme.

Single-group schemes

In this scheme, keys are created by an initiator of the group or by some centralized physical Group Controller (GC) and then delivered to the group. Each user u_i stores a group key g and a unique key encryption key k_i. The GC stores the group key as well as a single key l that is used to determine each users secret key $k_i = h_l(i)$. At a predetermined time all the participants switch to the group key, which is updated by the GC, after a predefined fixed time interval. During this time interval if a user joins the group a new group key is multicast to the group using the old group key and is privately sent to the new user using the user's private key k_i that is assigned by the GC. If a user leaves the group the GC sends out a new group key to the rest of the group using their individual keys. The distribution of the group key by the GC is *O(1)* during a join and *O(n)* during a leave where n is the number of users. The *O(n)* complexity during a leave can be reduced to *O(1)* in which case the participants will have to extract their keys from a single message from the root [13]. Figure 1 shows a diagram of a single-group multicast group with a GC and users *a* through *i*.

Group-of-groups schemes

This scheme is an extension of the single-group scheme. The group controller generates a group key g with the members of the group or without them. The GC transmits the key to the other members using each user's secret key k_i. The group is divided into subgroups to reduce the amount of packet transmissions sent out by the GC. Intermediary Group Controllers (IGCs) are chosen from each subgroup to perform the function of the GC for each local group. This scheme scales well but a lot of work is needed when an IGC leaves the group. This key distribution architecture is implemented in [13, 14, 15, 16, 18]. In [14] the key distribution is based on core based tree multicast routing where the GC is associated with the core router and the IGCs the other routers in the tree. In Iolus scheme [16] each subgroup has in addition its own local group key. In the scheme by Hardjono et al. [17] the group key is common to the members of the entire group while the subgroups have different intra-region group key as well as key management protocols. In this scheme revocation is performed by the local IGC who also picks a new group key for the entire group. The new group key is sent to the other IGCs who in turn multicast it to their respective local group. The number of communications during a revocation is $O(n_i)$ where n_i is the size of the local group in which the revocation occurs. Figure 2 shows this group with controller GC, intermediary controllers *A, B, C, D* and *E*, and group members *a, ...,* σ.

Single-tree schemes

This scheme attempts to provide secure removal of a compromised user from a multicast group with some transmission and storage

Figure 1: A Multicast Group with a Single Server (KDC)

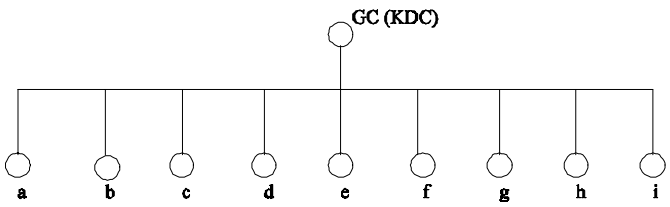
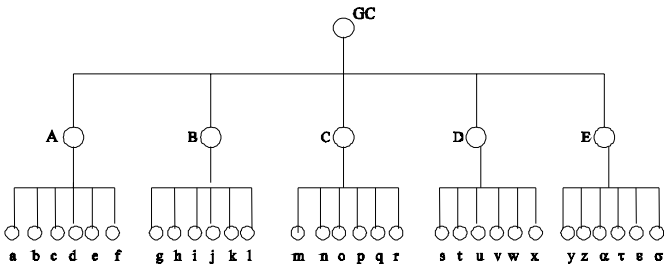


Figure 2: A Group of Multicast Groups.



efficiency. The scheme is introduced by Wallner et al. [13] with some extension in [18]. A rooted tree is created in which each leaf corresponds to a user each of whom has a unique key with the root (the group controller). The GC generates keys for every node in the tree and each user stores the keys of the nodes along the path connecting the user to the root. The keys stored by each user are transmitted to them through some secure channel. It is also possible to transmit all the keys in one message to the user and leaving the user to retrieve the node keys from the message. The root key becomes the group key since all the users in the group have it. The number of keys cached by each user is $\log_m n+1$ where n is the number of users and m the degree of the tree. The GC stores all the keys in the tree which is $(nm-1)/(m-1)$ for a full m-tree.

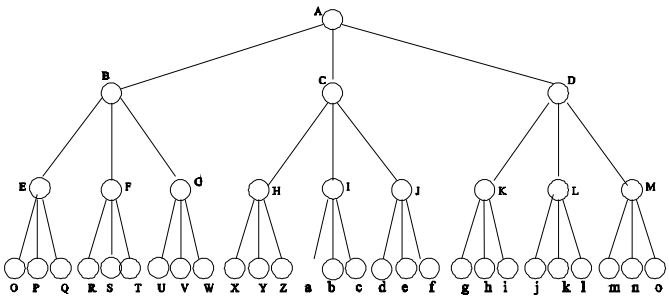
When a user joins the group the GC first multicast a new group key to the group then adds a new node to the tree for the new user. The GC sends the new group key and the keys for all the nodes along the path from the new user to the root, to the new user.

When a user leaves the group, the GC generates new keys for all the nodes whose keys the user possesses. Each node key is multicast to the sub-tree to which it is the root. The number of message transmissions required during a key revocation is $md \cong m\log_m n$ for an m-nary tree of depth d with n users. The minimum message transmission occurs at m = e = 2.71828. Since m is an integer the minimum occurs when m = 3. The number of required messages transmissions can be given as $O(\log_m n)$. Figure 3 shows a 3-nary tree with root *A*, intermediary nodes *B, C, ..., M*, and user nodes *O, P, ..., o*.

Group-of-trees scheme

In this scheme, the nodes corresponding to each user in the single-group scheme are replaced by the root nodes of single-tree schemes. The root nodes become the IGCs of their local single-tree schemes. We proposed that each local group have its own local session key g_i, which the IGC uses to multicast to the local group. Each IGC also stores a key IGC_{key} which it uses to multicast to the other IGCs and an overall multicast session key g. Let the i^{th} subgroup be made up of n_i users and m_i be the degree of the tree in the subgroup. As a consequence each user in the local group stores $\log_{mi} n_i + 2$ keys and the IGC stores $(m_i n_i-1)/(m_i-1) + 2$ keys for a full tree.

Figure 3: A Single Tree of Users.



During a user revocation, the local IGC creates a new overall session key g for the entire multicast group. The IGC uses the single-tree protocol to update g_i for its local group and then multicasts g to the local group. The local IGC then multicast g to the other IGCs using the multicast key IGC_{key}. The IGCs in turn multicast the new session key g to their local groups. The number of message transmission by the local IGC is $m_i d_i + 1$ where d_i is the height of the *m-nary* tree in the local group. The addition of a new user is done by the local IGC. The local IGC creates and distributes a new session key g for the whole group by first using the single-tree scheme to update the local session key g_i for the local group and later g. The local IGC multicasts g to the other IGCs which in turn multicast it to their respective local groups.

Tree-of-groups scheme

In this scheme, each leaf node in a single-tree hierarchy scheme is replaced by a single-group of n_i users. The leaves in the single-tree scheme correspond to IGCs. Each IGC has its local group key g_i and a key IGC_{key} used to multicast to the other IGCs. During a user revocation a similar protocol used in single-group scheme is used to update the local g_i and then g for the local group. The local IGC uses the IGC_{key} to multicast g to the other IGCs which is then multicast to their respective local groups. The number of message transmission by the local IGC during a revocation is $n_i + 1$. The ICG stores $n_i + 2$ keys whereas each user stores 3 keys.

Tree-of-trees scheme

The leaves in a single-tree scheme become the root of other single-tree schemes and IGCs of the multicast group. The overall tree can be considered as a large single-tree. The large single-tree can be unbalanced. The IGCs stores the keys g_i for each local group, the key IGC_{key} which is used to multicast to the other IGCs and the $(m_i n_i-1)/(m_i-1)$ keys for the nodes in the local tree. Each user stores $\log_{mi} n_i + 2$ keys.

A revocation is handled in a similar way as in the methods above. The local IGC updates the g_i and g for its local group and multicasts g to the rest of IGCs. The other IGCs in turn multicast g to their respective local groups. The number of transmissions is $m_i \log_{mi} n_i + 1$. The level in the large single-tree at which the nodes are considered ICGs could be chosen so as to minimize the amount of transmissions by the IGCs during a revocation and the storage size at both the ICGs and the users.

4. CONCLUSION

In this paper we identified various fundamental issues related to secure multicasting and the factors influencing the design of IP multicast security key distribution schemes which include the group dynamics and size, multicast application type, security policies, and trust with entities that generate, distribute and manage the cryptographic keys. The most important issue we have discussed in this paper is group key management schemes. We have expanded the two main existing schemes for key management, namely the single-group scheme and the single-tree scheme, by introducing three new schemes: group-of-trees, tree-of-groups and tree-of-trees. These new schemes are intended to handle multicast groups with very large number of users. We have formulated the number of messages and storage requirements of all these schemes in terms of the group size. However, our theoretical complexity analysis were derived under the idealistic assumption of having the distribution trees full and balanced which may not reflect the actual practical dynamics of group formations. A detailed simulation-based model and performance measurements are required to offer more realistic and practical results of our proposed scalable schemes.

REFERENCES

[1] Stinson, D. (1995). **Cryptography Theory and Practice**, CRC Press.

[2] Ghanem, S.M. and Abdel-Wahab, H. A. (2000). Simple XOR-based Technique for Distributing Group Key in Secure Multicasting, Technical Report, Dept. of CS ODU.

[3] Stallings, W. (2000). **Network Security Essentials Applications and Standards,** Prentice Hall.

[4] Deering, S.E. (August 1989). Host extensions for IP multicasting, RFC-1112.

[5] Handlley, M. (November 1996). Session Anoucement Protocol, Internet Draft.

[6] Handlley, M., Schulzrinne, and Schooler, (July 1997). Session Initiation Protocol, Internet Draft.

[7] Artkinson, R. (August 1995). Security Architecture for the Internet Protocol. RFC-1825.

[8] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L. (1997). A Reliable Multicast Framework for Light-weight Sessions and Application Level Framming, **IEEE/ACM Transactions on Networking, 5**(6), pp. 784-803.

[9] Paul, S., Sabani, K.K., Liu, J.C., and Bhattacharyya, S. (1997). Reliable Multicast Transport Protocol, **IEEE Journal on Selected Areas in Communications, 5**(3), pp. 407-421.

[10] Deering, S., Partridge, C., and Waitzman, D. (November 1988). Distance Vector Routing Protocol, RFC-1075.

[11] Balladie, T. (May 1997). Core Based Tres (CBT) Multicast Routing Architecture, RFC-.

[12] Moy, J. (March 1994). Multicast Extension to OSPF, RFC 1584.

[13] Wallner, D., Harder, E., and Agee, R. (July 1997). Key Management for Multicast: Issues and Architecture, Internet Draft, draft-wallner-key-arch-00.txt.

[14] Balladie, T. (1996). Scalable Multicast Key Distribution, RFC1949.

[15] Harney, H. and Muckenhirn, C. (July 1997). Group Key Management Protocol (GKMP) Architecture, RFC2094.

[16] Mittra, S. (1997). The Iolus Framework for scalable Secure Multicasting, **Proceedings of ACM SIGCOMM'97**.

[17] Hardjono, T., Cain, B.B., and Doraswamy, N. (July 1998). A Framework for Group Key Management for Multicast Security, Internet draft. draft-ietf-ipsec-gkmframework-00.txt.

[18] Wong, C.K., Gouda, M., and Lam, S. (1998). Secure Group Communication Using Key Graphs, **Proceedings of SIGCOMM'98.**