99-01

1999 EMERGING TECHNOLOGIES SYMPOSIUM

WIRELESS COMMUNICATIONS and SYSTEMS

APRIL 12 - 13, 1999

presented by:

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
DALLAS SECTION & REGION 5

OPTIMAL SCALABLE SECURITY ARCHITECTURES IN THE PRESENCE OF COLLUDING MOBILE TRAITORS

Radha Poovendran, John S. Baras

Department of Electrical Engineering &
Center for Satellite and Hybrid Communication Networks
University of Maryland, College Park MD 20742
e-mail: {radha, baras}@isr.umd.edu

Abstract – We present the issues related to secure multicast communication, in the presence of members who may collaborate to compromise the integrity of the system's security. We also show that the ability to compartmentalize the system compromise depends on the availability of trusted intermediate nodes. We also note that some variations of the recently proposed tree-based schemes don't provide the required level of security and may be compromised if two appropriate members are compromised. We present the analysis of the weakness of these schemes here. We further note that the currently available tree based key distribution schemes are not optimal, and choose the worst case solution for key assignment. We note that the claims, including the collusion, can be formally proved using basic concepts from source coding theory and entropy as in [4].

1. INTRODUCTION

Commercial applications such as pay per view, selective Internet stock quotes, Internet news and multimedia are examples of areas which will benefit from secure multicast communication. Unlike the unicast, which involves a pair of members, secure multicast may involve parties not only distributed but also at different integrity level. Since it is important to provide a common key for all the members in a secure multicast, any potential multicast key management scheme need to take into account several factors that are unique to multicast. Some of the important but yet unsolved challenges are (a) dealing with the dynamic nature of the group membership, (b) providing transparency to the key updates, and (c) dealing with the member collusion. We now describe the problems posed by these issues.

If the secure multicast group needs to protect the backward traffic information from the new joins, it must change its current traffic-encrypting key. For example, applications such as secure distributed conferences where the current members may not want to reveal the content of the past conversation to a new join have to change the keys. In order to be able to change the current traffic-encrypting keys, all the valid members must have a set of key-encrypting keys that can be used to update the traffic-encrypting keys whenever needed. Hence, at least at the level of the key distribution, problem of backward secrecy reduces to the problem of (a) choosing an appropriate traffic-encrypting key and (b) providing a mechanism for all the members to securely update the keys. The following two encryption steps can achieve this: (a) use a common key-encrypting key of the current users and encrypt the new traffic key using the common key encrypting key and (b) use an individual key encrypting key to securely provide the new member with the new traffic key. This approach works well if there is no member *revocation or leave*. In the event that one or more members have to leave the session in the middle, protection of the forward or future traffic becomes critical to preserve the integrity of the communication. In such a situation, relying on a single key-encrypting key doesn't preserve the forward secrecy. Hence, clever key distribution techniques, which use more than one key-encrypting key to be shared by the group members to minimize the required encryption at the time of key update due to (a) member join and (b) member leave/revocation are needed.

Another major problem that is unique to group communication is user collusion. Under user collusion, two or more members may cooperate and possibly compute the key-encrypting keys or other secret parameters of a non-colluding member. Providing collusion resistant key distribution schemes at lower

0-7803-5554-7/99/$10.00 © IEEE.

computational and storage burden is an important problem in secure multicast communication as well. We will later show that two of the recently proposed secure multicast schemes have very serious problems under user collusion. In fact, we show in this paper that the integrity of these schemes can be broken if two appropriate members were to collude.

The paper is organized as follows: Section 2 presents a summary of the recent key distribution schemes for secure multicast and focuses on the tree based schemes that have lesser encryption overheads. Section 3 presents the user collusion problem with a special case of tree-based key distribution results reported in [2, 3]. Section 4 shows that the current tree based schemes assume the worst case scenario at the time of design and also assume the total membership is known in advance. Using these arguments, we show that the storage requirements at the group center level as well as the user level are the highest under the current tree based schemes.

2. KEY DISTRIBUTION METHODS

Since all the members of the multicast group have to share the same traffic-encrypting key, one simple approach is to distribute the same key-encrypting key to all the members. At the time of traffic key update, such an approach would require single encryption at the group center and one decryption at each user node. In this scheme, if a single member is compromised, the whole group will loose the forward secrecy. Any future key update will be exposed to the attacker and hence the system integrity doesn't *scale* beyond one member. Once a single node is compromised, the group center has to either live with the compromised member or regroup. A variation of this key distribution approach is based on the observation that a large group can be compartmentalized into clusters with their own key encrypting keys and the traffic encrypting keys. Such an approach can be viewed as a compartmentalization of a heterogeneous group into homogeneous subgroups. By clustering the group, the amount of required encryption is restricted to the size of the cluster. Since each cluster has its own keys, this approach of key distribution involves additional encryption or decryption overhead if the clusters want to communicate.

Another approach is to assign unique keys to individual members and to update the traffic keys only for the valid members. This approach involves *O(N)* encryption at the group center and one decryption per member at the user level.

Recently an approach based on trees was proposed [6, 7]. Figure 1 presents a rooted tree based hierarchical key management approach called Logical Key Tree (LKT) or key graph. In this approach, each leaf represents a unique member of the group. *Each* node in LKT represents a key, with $\mathbf{K_O}$ being the common key for the whole group. Since the member and the leaf indices are mapped in a one-to-one manner, it is possible to assign a unique set of keys for a given member. The set of keys held by a member is the set generated by traversing the tree from the root to the leaf with inclusion of the root and the leaf keys. For example, $\{\mathbf{K_O}, \mathbf{K_M}, \mathbf{K_I}, \mathbf{K_A}, \mathbf{K_1}\}$ represents the unique set of keys held by member 1. Hence, the group center can use a lookup table consisting of the member index and the corresponding set of keys for revocation. Since key $\mathbf{K_O}$ is common to all the members, it need not be stored against each index.

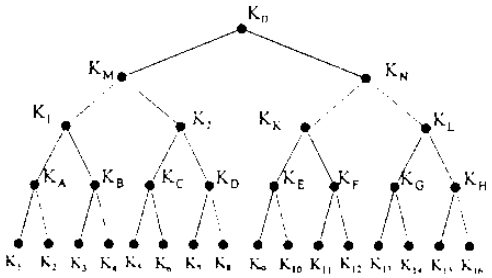

Figure 1. Logical Key Tree with leafs representing users

For a rooted tree of depth h, each member stores $(h+1)$ keys, h of which will be shared with one or more members. We now describe the member revocation in a rooted tree framework.

2.1 Member Revocation in Rooted Trees

In the event of key updates due to a member revocation event, the unique mapping between the member index and the set of keys given to a member helps to update only the ``compromised keys". We note that the root key and the traffic key are to be updated for every membership change

Member 1 in Figure 1 is indexed by the set of five keys $\{K_O, K_M, K_I, K_A, K_1\}$. In this structure, the whole group can update its keys in *O(log N)* encryption at the group center and one decryption at each leaf node. For example, if member 1 is to be revoked, keys $\{K_O, K_M, K_I, K_A, K_1\}$ need to be updated for relevant members. The following set of members require different subsets of the keys held by member 1: (a) members 9-16 need to update K_O, (b) members 5-8 need to update $\{K_O, K_M\}$, (c) members 3-4 need to update $\{K_O, K_M, K_I\}$, and (d) the member 2 needs to update $\{K_O, K_M, K_I, K_A, K_1\}$. The following lemma summarizes the number of keys to be updated under member revocation.

Lemma 1. Revocation of a single member in a binary tree with N members requires *(1+ logN)* keys to be updated.

Proof: Direct result of the fact that the total keys to be updated include all the keys from the root to the leaf on the tree and the traffic encrypting key, excluding the compromised leaf key.

3. WEAKNESS OF SOME TREE BASED SCHEMES

We note that the currently available tree based schemes assume that the group size is known in advance, and design a full binary tree of depth *logN*. If the tree is full, and has depth *h*, then the total number of different keys that need to be stored at the group center is $(2^{h+1} - 1)$. A recent approach proposed to assign the key based on a unique id assignment. In this method [8,9], authors note that for a group size N, there need to be *logN* bits to uniquely index each member. Since each bit takes two values, if a pair of unique keys is assigned to represent the bit location, *2logN* key are sufficient to uniquely index N members.

Bit #	Key value for 0	Key value for 1
1	K_1.0	K_1.1
2	K_2.0	K_2.1
3	K_3.0	K_3.1
4	K_4.0	K_4.1

Table 1: Member id-key mapping

Table 1 presents the id based key allocation approach described above for a group size of 16. In this case, four bits are needed to uniquely index each of the 16 members. In the table, bit # is counted from left to right in an id representation, although order of assignment does not matter. From the table, it looks like this scheme has reduced the storage overhead by a significant factor and is able to reduce the needed key updates in the event of a member revocation. However, a careful look at the approach reveals this method of key assignment suffers from collusion among members and *under collusion, it does not scale beyond 4 members!* In order to demonstrate the problem associated with this scheme, we present the tree version of the scheme.

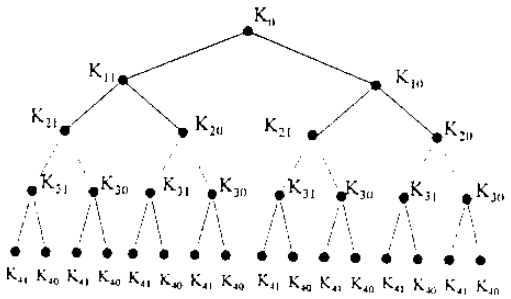

Figure 2. Member id-key Tree

3.1 User Collusion

We define user collusion as: *Given a set of members with keys that are not necessarily unique, if two or more members can cooperate and construct a set of keys that include all keys of another member, the system is said to be susceptible to user collusion.*

In the current id based key scheme, if members 1 and 16 collude, all eight keys of the group will be exposed. Hence, the integrity of the group will be fully compromised. In fact, for a binary tree, $N > log\ N$, if $N > 4$. In such case, there is no key assignment that will prevent user collusion since there is no way to assign unique key to an individual member. Hence, such a tree scheme does not scale beyond two members under collusion. In other words, *for a full binary tree based key assignment, independent of how large the group size is, it takes only two members to compromise the whole group integrity as long as the keys are assigned using id based approach.*

From Figure 2, we also note that more than one pair of members may be able to collude in compromising the group communication.

4. OPTIMALITY OF THE CURRENT SCHEME

We note that if we define the event "member revocation" and assign probabilities to each member, we can show that the full trees such as the ones reported in the recent literature assume that all the members have equal probability of being revoked. Under such condition, the probability of member revocation is uniform. Moreover, since each member is uniquely assigned to a leaf node, the probability of member revocation is also the probability of the leaf key revocation. Hence the information theoretic entropy of member revocation is the same as the entropy of the leaf key revocation. This result is not surprising since the leaf keys are supposed to be unique for each member. If the individual member revocation events are independent, then the revocation probability of any intermediate node is the sum of the revocation probabilities of the descendant leafs. We note without proofs that the uniform member revocation probabilities can be tied to the optimal number of keys a member has to be given for collusion free key assignment. Under this condition, it can be shown that the tree based key assignments lead to maximum entropy of the member revocation event. Since the *average number of keys per member is related to the entropy of member revocation, using tree solution leads to the largest possible optimal key assignment.* Details of the proofs are beyond the scope of this paper and are discussed in [4].

5. CONCLUSIONS AND OPEN PROBLEMS

This paper studied the recently proposed tree based secure multicast key distribution schemes and presented the problems associated with some variations of it. The paper also showed that collusion resistance is equivalent to having a unique key assignment for each member. We also noted that the entropy of member revocation is the same as the entropy of the leaf key revocation and the average number of keys held per person is equal to the entropy of member revocation. Analytical derivations are based on source coding results of information theory. We note that although the source coding results are applicable for optimal tree design, use of optimal coding in key assignment will lead to collusion problems. The analysis of collusion from the point of optimal codes is presented in [4]. One of the issues that need to be addressed in the literature is the optimal security parameter selection under mobility. This problem relates the mobility and the security requirements to the available hardware.

REFRENCES

(1) Canetti, R., Pinkas, B. ``A taxonomy of multicast security issues", Internet draft, November 1998.

(2) Steiner, M., Tsudik, Waidner, M., ``Diffie-Hellman key distribution extended to group communication", 3rd ACM Conference on Computer and Communications Security, 1996.

(3) Menezes, A., van Oorschot, A., Vanstone, A., *Handbook of Applied Cryptography*, CRC Press, Boca Raton, 1997.

(4) R. Poovendran, Baras, J. S., ``An Information Theoretic Approach to Secure Multicast Key Management", Proceedings of IEEE Information Theory and Networking workshop, Metsovo, Greece, June 1999.

(5) M. Luby, *Pseudo-Random Functions and Applications*, Princeton University Press, 1996.

(6) Wallner, D. M., Harder, E., C., Agee, R. C., ``Key Management for Multicast: Issues and Architectures", Internet draft, September 1998.

(7) Wong, C. K., Gouda, M., Lam, S. S., ``Secure Group Communications Using Key Graphs", In Proceedings of ACM SIGCOMM'98, September 2-4, Vancouver, Canada.

(8) Caronni, G., Waldvogel, M., Sun, D., Plattner, G., ``Efficient Security for Large and Dynamic Groups", In Proceedings of the Seventh Workshop on Enabling Technologies, IEEE Computer Society Press, 1998.

(9) Chang, I., Engel, R., Kandlur, D., Pendarakis, D., Saha, D., ``Key Management for Secure Internet Multicast Using Boolean Function Minimization Techniques", Proceedings of IEEE Infocom'99.

(10) Mittra, S., ``IoLus: A framework for Scalable Secure Multicasting", in Proceedings of ACM SIGGCOM'97, pages 277--288, September 1997.

(11) Harney, H., Muckenhirn, ``GKMP Architecture", Internet RFC 2093, July 1997.

(12) Canetti, R., Cheng, P. C., Pendarakis, D., Rao, J. R., Rohatgi, P., Saha, D., ``An Architecture for Secure Internet Multicast", Internet Draft, November 1998.

(13) Hardjono, T., Cain, B., Doraswamy, N., ``A Framework for Group Key Management for Multicast Security", Internet draft, July 1998.

(14) Quinn, B., ``IP Multicast Applications: Challenges and Solutions", Internet draft, November 1998.

(15) Ballardie, A., ``Scalable Multicast Key Distribution", Internet RFC 1949, May 1996.