I.J. Information Technology and Computer Science, **2012, 1, 32-39**
Published Online February 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.01.05

A New Secure Multicast Key Distribution Scheme Using Tabulation Method

R. Varalakshmi
Teaching Research Associate – Department of Mathematics,
Anna University, Chennai, India
Email: rvaralakshmi@annauniv.edu

Dr. V. Rhymend Uthariaraj
Professor and Director, Ramanujan Computing Centre
Anna University, Chennai, India

***Abstract*— In the present paper, we propose a new scheme for a scalable multicast key distribution scheme. The present scheme is based on the Key Management using Tabulation method of Boolean Function Simplification technique. It explores the use of batching of group membership changes to reduce the frequency, and hence the cost, of key re-distribution operations. It focuses explicitly on the issue of snowballing member removal and presents an algorithm that minimizes the number of messages required to distribute new keys to the remaining group members. The algorithm is used in conjunction with a new scalable multicast key distribution scheme which uses a set of auxiliary keys in order to improve scalability. In contrast to previous schemes which generate a fixed hierarchy of keys, the proposed scheme dynamically generates the most suitable key hierarchy by composing different keys. Our snowballing member removal uses one of the Boolean function simplification techniques called tabulation method, and outperforms all other schemes known to us in terms of message complexity. Most importantly, our technique is superior in minimizing the number of messages when multiple members leave the session in the same round.**

***Index Terms*— Multicast key distribution, Snowballing member removal, Boolean Function Simplification, tabulation method, Communication overhead.**

Manuscript received July 13, 2011; revised December 19, 2011; accepted December 31, 2011.

1. Introduction

Many applications like pay-per–view, distribution of digital media etc., require secure multicast services in order to restrict group membership and enforce accountability of group members. A major issue associated with the deployment of secure multicast delivery services is the scalability of the key distribution scheme. This is particularly true with regard to the handling of group membership changes, such as membership departures and/or expulsions, which necessitate the distribution of a new session key to all the remaining group members.

As the frequency of group membership change increases, it becomes necessary to reduce the cost of key distribution operations. One solution is to let all authorized members use a shared key to encrypt the multicast data. To provide backward and forward confidentiality (D.M. Wallner and Agee, 1999), this shared key has to be updated on every membership change and redistributed to all authorized members securely which is referred to as rekeying. The efficiency of rekeying is an important issue in secure multicast as this is the most frequently performed activity with dynamic change in the membership.

Group key must be updated with the group membership changes to prevent a new member from deciphering messages exchanged before it join the group; this is defined as backward secrecy [3]. Group key

revocation in case of one member joins or multiple members join could be achieved by sending the new group key to the old group members encrypted with the old group key. Also, group key must be must be updated with the group membership changes to prevent an old member (leaved or expelled) from deciphering current and future communication which is defined as forward secrecy[3]. Group key revocation, when one member leaves or multiple members leave, is more complicated in case of join because of the disclosure of the old group key. The old group key is known to the leaving member(s) so there is a need to re-key the group using valid key(s) in a scalable way. The trivial scheme for rekeying a group of n members is through using individual secret key shared between the Key distribution Centre KDC and each member. This is not a simple or scalable method and consumed large bandwidth especially for large group with high membership changes: furthermore it takes more time and needs more resources per hosts than using multicasting to re-key the group.

The rest of the paper is organized as follows. Section II discusses related work. The multicast key management scheme is presented in Section III. Section IV describes the algorithm that uses the Boolean function simplification techniques called tabulation method to minimize the number of rekeying operations. Section V analyzes the performance of the proposed scheme, followed by the conclusion in Section VI.

2. Related Works

The topics of key management for multiparty communications in general networks are studied and one of the efficient key tree based group key management technique called Logical Key Hierarchy (LKH) is discussed [1,2,3,4,5,6]. A key update in this scheme requires $O(\log_2 N)$ messages where N is the size of the group. In this scheme each user has to store $\log_2 N$ keys (i.e., keys along the path from leaf to the root) and the key server has to maintain a tree of O (N) keys.

The scheme proposed in uses the LKH scheme and uses a binary tree, but with only two keys at every level.

This reduces total number of keys at the server from O(N) to O(h) where h is the height of the tree. But storage at each user remains at $O(\log_2 N)$. The scheme discussed and extends the scheme to m-ary tree instead of binary tree, which reduces the user side storage from $O(\log_2 N)$ as to $O(\log_m N)$[2]. In tree based key management schemes each user shares a key called private key with the key server and key at the root of the tree is the group key which is shared by all users in the group. Other keys (other than private key and group key) are called auxiliary keys (key encryption keys) which are known only for certain subset of users and are used to encrypt new group key whenever there is a group membership change.

The scheme uses m-ary tree and at each level m keys are maintained. Whenever a node is compromised new group key is selected and distributed to other nodes. The encryption keys that are required to send new group key GKnew securely are computed. The new group key is distributed to group members without performing any encryptions. Our scheme distributes new group key to the remaining group members with minimum number of messages as compared to the scheme in [7]. In our scheme, in order to avoid the leaving members using auxiliary keys to learn the new group key, auxiliary keys are also updated.

3. Multicast Key Management Scheme

In our scheme each member of the group is associated with a unique user ID (UID) which is a binary string of length n. Consequently, a UID can be written as $X_{n-1}\ X_{n-2}\ X_0$, where X_i can be either 0 or 1. Using Boolean notation, X_i / can be written as x'i or xi / depending on whether ! is Xi is 0 or 1. The length of the UID depends upon the size of the multicast group.

In [3] binary tree structure is used. When the group is large, the number of levels in the binary tree will be more which increases number of keys at member. Extending the scheme to m-ary tree will reduce the height of the tree reducing number of keys at each member. At the same time we should consider server side storage i.e., number of keys at every level of the key tree.

In [3] two keys are maintained at every level of the key tree, extending the scheme to m-ary tree will result in maintaining m keys. For a group size n, if d is the height of the binary tree, it results in storing 2*d keys at the server. For the same value of n, if d' is the height of the m-ary tree, then m*d' keys are to be stored at the server. We can have the relation

$n = 2d = md'$

$\rightarrow d' = d/\log_2 m$

Number of keys at server in m-ary tree in terms of d can be represented as $m*(d/\log_2 m)$, which illustrates that as m increases, number of keys at server will increase, which violates our motto. Hence in order to maintain minimum number of keys both at member and server, following relation has to be satisfied :

$(m*d/\log_2 m) \leq 2*d$ which is true only if $m \leq 4$.

Notations

m-ary tree: is a tree with the following properties:

(i) each interior node has at most m children

(ii) each path from the root to a leaf has the same length

N: Total number of members associated with the group. Each member is assigned with Unique Identification Number (UID) which is a binary string of length n (where $n = \log_2 N$).

Subgroups: Each interior node containing at the maximum m children nodes forms one subgroup.

Subgroups at level i is where the leaders reside and are assigned with keys Ki0 to K i(m-1) called Auxiliary keys at level i.

Keys: Individual member keys of any subgroup are numbered from K_0 to K_{m-1} so that the leaders at level i are assigned with key K_0 and members at position 1 of all subgroups are assigned with key K_1 and all members at position 2 of all subgroups are assigned with key K_2 and so on up to K_{m-1}.

KEK: Key Encryption Keys is the set, initially empty, and at the end contains the keys used to encrypt the new auxiliary keys and member keys.

$\{GK\}\ K_1$ denotes GK is encrypted with the key K_1.

|| denotes concatenation operation

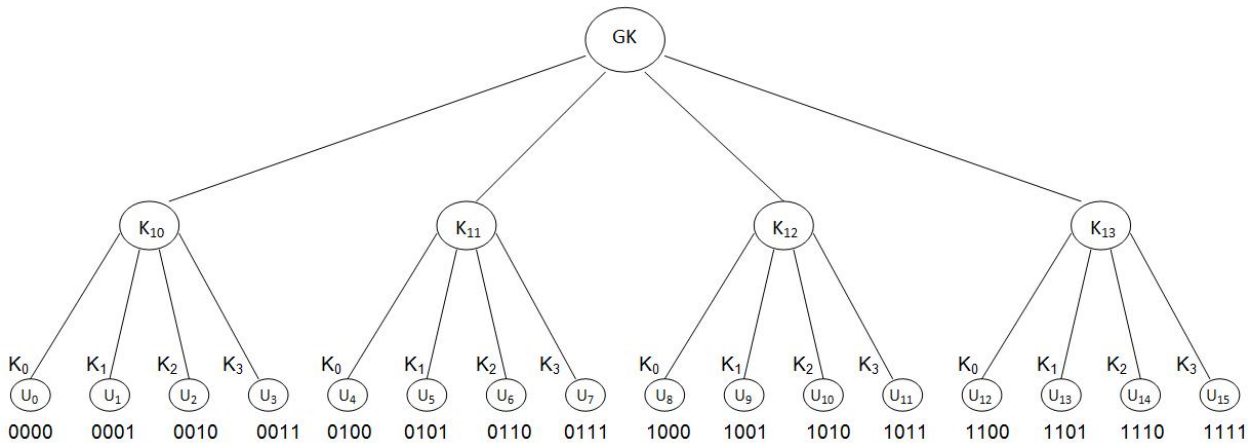

Fig 1: Key tree structure showing UIDs and keys of users in the group, auxiliary keys and group key

From fig. 1 the values of N, m, n, keys, auxiliary keys and group key are as follows:

N=16 m=4 n=4

Keys:

Members u_0, u_4, u_8, u_{12} are assigned with key K_0

Members u_1, u_5, u_9, u_{13} are assigned with key K_1

Members u_2, u_6, u_{10}, u_{14} are assigned with key K_2

Members u_3, u_7, u_{11}, u_{15} are assigned with key K_3

K_{10}, K_{11}, K_{12}, K_{13} are auxiliary keys at level 1.

GK is the group key shared by u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}

GKnew : new group key

3.1 Procedure for finding new Secret Key

Using Hash function the method used to communicate the new secret key is as follows: central node computes the hash ([9, 10] of a shared key (i.e., key known to central node and authorized node) say ks i.e., H(ks) of the encryption key and XOR's with new secret key knew to be communicated.

k comm $\leftarrow$ H(ks) $\oplus$ knew

After getting k comm nodes having ks compute H(ks) and XOR's with k comm which yields new secret key knew.

i.e., knew H(ks) $\oplus$ k comm

3.2 Key Distribution

The encryption keys computed using the method of [11] are used to communicate new group key to the existing nodes without actually performing any encryption. Messages send by central node to group members by using the hash of the encryption keys that are known to compromised nodes. Hence using the keys of the compromised nodes it is not possible to get any information regarding new group key. In order to avoid attackers decrypting any message in the next time interval we perform two operations. First, each remaining node along with path from the leaving point will compute new auxiliary key using the method, F (auxiliary key, new group key)

(auxiliary key) $\oplus$ (new group key). Second, every key used to compute the hash value is incremented by one (1). In this scheme to communicate new group key securely we are not using any encryption instead all communications are by using hash values and XOR operations which will reduce the communication overhead i.e., rekeying cost is reduced.

3.3 Individual Member Removal

When a member of a multicast group is to be expelled, e.g., because its subscription has expired, a new session key needs to be distributed to every member except the one leaving to make sure that the expelled member can no longer receive and send data addressed to the group. Similarly, if a member voluntarily leaves the multicast group, the session key might also have to be updated. This can be useful for sessions where members pay according to the duration of their membership in the group.

In fig.1 if member u_1 leaves, the Controller generates new keys and conveys new keys to the remaining members through a set of rekeying messages as:

KEK = {{ GKnew } K_0 || GKnew } K_1 || { GKnew } K_3 || { GKnew } K_{11} || { GKnew } K_{12} || { GKnew } K_{13} } }

After distributing new keys to remaining members in the multicast group securely, auxiliary keys are updated using the function F as follows:

F(auxiliary keys, new group key) <- auxiliary keys $\oplus$group key

Same method holds good for all the following cases to compute new auxiliary keys.

3.4 Two members removal (leave) at a time

When two members leave the multicast group voluntarily or being removed from the group, we need to address three different cases:

(i) both the leaving members are from the same subgroup,

(ii) leaving members belonging to different subgroups but at the same position with common individual key (for example, in figure 1 u_1 and u_5 are the members belonging to different subgroups sharing the key K_1),

(iii) leaving members belonging to different subgroups and also at different positions with different individual keys (for e.g., in fig.1 users u_6 and u_{11} belong to subgroup 2 and 3 respectively with individual keys being K_2 and K_3 respectively).

Case (i): Let Leaving members be u_5 and u_6

/* leaving members from the same subgroup */

KEK= { K_{10}, K_{12}, K_{13}, K_0, K_3 }

Following users can decrypt the new key encrypted using the keys of set KEK:

u_0, u_1, u_2, u_3 (using key K_{10})

u_8, u_9 ,u_{10}, u_{11} (using key K_{12})

u_{12}, u_{13}, u_{14} , u_{15} (using key K_{13})

u_4 (using key K_0)

u_7 (using key K_3)

For the same members removal, Wong et al. scheme of [2] requires 6 encryptions whereas our scheme requires only 12 rekeying operations and 5 encryptions.

Case (ii): Let Leaving members be u1 and u9

/* leaving members are from the same position of different subgroups */

KEK = { K_{11}, K_{13}, K_2 , K_0, K_3}

Following users can decrypt the new key encrypted using the keys of set KEK:

u_4, u_5, u_6, u_7 (using key K_{11})

u_{12}, u_{13} ,u_{14}, u_{15} (using key K_{13})

u_0, u_4 ,u_8, u_{12} (using key K_0)

u_2, u_6, u_{10}, u_{14} (using key K_2)

u_3, u_7 ,u_{11}, u_{15} (using key K_3)

For the same members removal, Wong et al. scheme of [2] requires 10 encryptions, whereas our scheme requires only 5 encryptions.

Case (iii): Let Leaving members u_2 and u_{13}

/* leaving members are from different positions of two different subgroups */

KEK = {K_0, K_3, K_{11}, K_{12}, $K_{13}\oplus K_2$, $K_{10}\oplus K_1$}

Following users can decrypt the new group key GKnew encrypted using the keys of set KEK:

u_0, u_4 ,u_8, u_{12} (using key K_0)

u_3, u_7 ,u_{11}, u_{15} (using key K_3)

u_4, u_5, u_6, u_7 (using key K_{11})

u_8, u_9, u_{10}, u_{11} (using key K_{12})

u_{14} (using key $K_{13} \oplus K_2$)

u_1 (using key $K_{10} \oplus K_1$)

For the same members removal, Wong et al. scheme of [2] requires 10 encryptions, where as our scheme requires only 6 encryptions.

4. TABULATION METHOD

4.1 Snowballing member removal

Any number of members can leave (be removed from) the multicast group from any position in our decentralized key management scheme. The Controller executes the tabulation method to compute the messages

that need to be sent out after multiple members depart the group in the same round.

There is a straightforward analogy between minimizing boolean functions and aggregating re-keying messages. Therefore, the need for simplification and aggregation arises. Similar problems have been addressed for many years in the area of switching theory and logical design. The objective there is to minimize Boolean functions so that the complexity of digital circuits can be reduced. In the context of logical design, a + operation corresponds to an OR gate and a multiplication to an AND gate. Typical objectives include the simplification of total number of gates and/or number of circuit stages.

We borrow from the results of logical design to construct a more efficient re-keying process. First, we define some of the terms we use in subsequent discussions.

Literal: A variable or its complement

Product Term: Series of literals related by AND

Minterm: A product term which contains as many literals as there are variables in the function.

Sum term: Series of literals related by OR

Maxterm: A sum term which contains as many literals as there are variables in the function.

Normal term: Product or sum term in which no variable appears more than once.

TABLE I Determination of Prime Implicants

	a	**b**	**c**	**d**
0	0	0	0	0
4	0	1	0	0
8	1	0	0	0
3	0	0	1	1
5	0	1	0	1
6	0	1	1	0
10	1	0	1	0
12	1	1	0	0
7	0	1	1	1
11	1	0	1	1
13	1	1	0	1
14	1	1	1	0

	a	**b**	**c**	**d**	
(0,4)	0	#	0	0	√
(0,8)	#	0	0	0	√
(4,5)	0	1	0	#	√
(4,6)	0	1	#	0	√
(4,12)	#	1	0	0	√
(8,10)	1	0	#	0	√
(8,12)	1	#	0	0	√
(3,7)	0	#	1	1	
(3,11)	#	0	1	1	
(5,7)	0	1	#	1	√
(5,13)	#	1	0	1	√
(6,7)	0	1	1	#	√
(6,14)	#	1	1	0	
(10,11)	1	0	1	#	√
(10,14)	1	#	1	0	
(12,13)	1	1	0	#	
(12,14)	1	1	#	0	

	a	**b**	**c**	**d**
(0,4,4,12)	#	#	0	0
(0,4,8,12)	#	#	0	0
(0,8,4,12)	#	#	0	0
(0,8,8,12)	#	#	0	0
(4,5,5,7)	0	1	#	#
(4,5,6,7)	0	1	#	#
(4,6,5,7)	0	1	#	#
(4,6,6,7)	0	1	#	#
(8,10,10,11)	1	0	#	#
(3,7,3,11)	#	#	1	1
(5,13)	#	1	0	1
(6,14,10,14)	#	#	1	0
(12,13,12,14)	1	1	#	#

The standard form most usually considered in the simplification of Boolean functions is the form known as the sum of products expression (SOPE). In the context of logical design, each product corresponds to an AND gate and each literal to a gate input. In the context of the multicast group re-keying problem, each product corresponds to a message and each literal to a key which is used as input to a function that derives the encryption/decryption key for the message. Many criteria can be applied in optimizing a sum of products form. In the context of logical design, a sum of products expression is regarded as a minimal expression if there exist(1) no other equivalent expression involving fewer products, and (2) no other expression involving the same number of products but a smaller number of literals.

The tabulation method is a specific step-by-step procedure that is guaranteed to produce a simplified standard-form expression for a function. It consists of two parts. The first is to find by an exhaustive search all the terms that are candidates for inclusion in the simplified function. These terms are called prime-implicants. The second operation is to choose among the prime-implicants those that give an expression with the least number of literals.

Illustrative example:

Let leaving members be u_1, u_2, u_9, u_{15}

Thus, f(a,b,c,d) = Σ (1,2,9,15)

Remaining members

f(a,b,c,d) = Σ (0,3-8,10-14)

F= a'b + c'd' + cd + bc'd + cd'

4.2 Determination of prime implicants

Table 1 describes the procedure to determine the prime implicants.

Successive procedure: (algorithmic description)

1. Determination of the Disjunctive Normal Form and a list of minterms
2. As far as possible: pairwise combination of (min)terms and set up of a list of products
3. Repetition of 1.2 with an updated list after every repetition
   until:
4. no further simplification is possible any more.

Result on termination: prime implicants of the function f

Let Leaving members u_1, u_2, u_9 and u_{15}

/* leaving members are from different positions of two different subgroups */

KEK = {K_0, K_{11}, $K_1 \oplus K_{13}$, $K_3 \oplus K_{12}$, $K_3 \oplus K_{10}$, $K_2 \oplus K_{12}$, $K_2 \oplus K_{13}$}

u_4, u_5, u_6, u_7 (using key K_{11})

u_0, u_8, u_{12} (using key K_0)

u_3 (using key $K_3 \oplus K_{10}$)

u_{10} (using key $K_2 \oplus K_{12}$)

u_{11} (using key $K_3 \oplus K_{12}$)

u_{13} (using key $K_1 \oplus K_{13}$)

u_{14} (using key $K_2 \oplus K_{13}$)

For the same members removal, Wong et al. scheme of [2] requires 10 encryptions, where as our scheme requires only 7 encryptions.

5. PERFORMANCE COMPARISON

To achieve message confidentiality in Secure Group Communication we require a group key and the group key should be updated whenever a node is compromised. In our scheme server is required to store ($\log_2 N$ * m) keys, along with the Group Key GK, where as the scheme in [7] requires O (N) keys to be stored at the server. The binary tree concept discussed in [8] is efficiently extended to m-ary tree in this paper with

reduced storage at user side. New Group Key is distributed to the existing nodes using hash functions and XOR operations.

6. CONCLUSION

The rationale behind the definition of optimality is that typically the cost of an additional gate is several times that of an additional input on an already existing gate and, hence, elimination of gates is the primary objective of the simplification process. Interestingly, the same definition of optimality is also applicable to our problem. The argument in our case is that the complexity of sending an additional message is far greater than that of adding an extra key ID in the message to indicate that the key should be used as input in deriving a new key.

The proposed scheme is based on KM-BFM scheme. Instead of using one tree as in KM-BFM, the members are divided into a number of subgroup trees. A comparison between the proposed scheme and KM-BFM scheme is undertaken according to storage requirements at both group controller and group members and the number of updates in case of a single leave or multiple leaves. The comparison shows that the proposed scheme using tabulation method achieves lower storage requirements at both the group controller and the group members. On the other hand, it has a lower communication overhead in case of a single member leave and a comparable communication overhead in case of multiple leaves.

References

[1] A.Bellardie, ”Scalable Multicast Key Distribution” RFC 1949, May 1996

[2] Chung Kei Wong, Mohamed Gouda, and Simon S Lam, “Secure Group Communication Using Key Graphs”, Proceedings of ACMSIGCOMM, Vancouver, British Columbia, September 1998.

[3] I. Chang, R.Engel, D.Kandlur, D.Pendarakis and D.Daha. “Key management for secure internet multicast using Boolean function minimization technique”. ACM SIGCOMM’99, March 1999.

[4] Debby M. Wallner, Eric J. Harder, Ryan C. Agee, “Key Management for Multicast: Issues and Architectures”, Informational RFC, draft-Wallnerkey-arch-ootxt, July 1997.

[5] H.Harney, C.Muckenhirn, “Group Key Management Protocol (GKMP) Architecture”, RFC 2094, July 1997.

[6] H.Harney, C.Muckenhirn, “Group Key Management Protocol (GKMP) Specifications”, RFC 2093, July 1997.

[7] D.McGrew and A. Sherman. “Key establishment in large dynamic groups using one way function trees”, May 1998.

[8] A. Perrig, D.Song and J.Tygar, “ELK: A new protocol for efficient large-group key distribution”. In Proceedings of the 2001 IEEE symposium on Security and Privacy, 2001.

[9] Ran Canetti, Benny Pinkas, “A Taxonomy of Multicast security issues”, Internet Draft, May 1998.

[10] Suvo Mittra, “Iolus: A Framework for Scalable Secure Multicasting”, Proceedings of ACMSIGCOMM’97, Cannes, France, pp. 277-288, 1997.

[11] A.Chandha, Y.Liu and S.K.Das, “Group Key distribution via local collaboration in wireless sensor networks”, in IEEE International Conference on Sensor and AdHoc Communications and Networks(SECON), 2006.

[12] Chaddoud, G., Chrisment, I. and Schaff, A., “Dynamic Group Communication Security”, Proc. of sixth IEEE Symposium on Computers and Communications (ISCC'01) pp 49-56, 2001.

[13] Chang I., Engel R., Kandlur D., Pendarakis, Dimitrios., Saha, Debanjan., “Key Management For Secure Internet Multicast Using Boolean Function Minimization Technique”, Proc. of INFOCOMM’99, pp 689-698, 1999.

[14] E.L.McCLUSKEY, Minimization of Boolean functions, Bell System Technical Journal, 35 (1959), pp. 149-175.

[15] Moyer, M.J., Rao, J.R and Rohatgi, P., “A Survey Of Security Issues in Multicast Communications”, IEEE Networks vol. 13, pp12-23, 1999.

[16] Roth, Charles H., Jr, “Fundamentals of Logic Design”, Fourth Edition, pp 161-167.

[17] Setia, S., Koussih, S., Jajodia, S., and Harder, E., “Kronos: “A Scalable Group Re-keying Approach For Secure Multicast”, Proc. of 2000 IEEE Symposium on Secusrity and Privacy, pp 215-228, 2000.

