INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2001; **11**: 365–374 (DOI: 10.1002/nem.412)

Secure key agreement for group communications

By Wen-Her Yang and Shiuh-Pyng Shieh*

A secure key agreement protocol for group communications is proposed in this paper, which ensures the authenticity of group members and the privacy of group messages, and provides the properties of perfect forward and backward privacy. In a group session, the common key is collaboratively established by all participants, hence the overhead of key agreement is balanced among group members. Copyright © 2001 John Wiley & Sons, Ltd.

Introduction

In recent years, many network applications reliant on group communications have been developed. It is important to ensure security in the group communication environment. For example, a group member may want to make sure that all participants are not impersonated and communication data are protected from eavesdropping. The basic requirement for secure group communications is that a common group key must be generated for all members to confidentially communicate with others. Generally, a common group key is established by a key establishment protocol. Group key establishment protocols can be classified in two categories: key agreement and key distribution. In key agreement protocols, group keys are generated collaboratively by all members who join the group. Each party contributes a secret value to construct a common key, which is known only by group members after the protocol is completed. On the other hand, key distribution protocols require a dedicated central server to be in charge of key generation and distribution. To simplify protocol design, key distribution by a central server is often adopted by many schemes.[1–5] However, centralized approaches may not suitable for many applications, where participants wish to generate group keys by themselves so that they can be sure of the freshness and randomness of group keys. In the centralized approach, the central server is likely becomes a performance bottleneck and a very attractive target for crackers. For these reasons, key distribution protocols have some limitations and are unsuitable for dynamic group communication.

Many key agreement protocols[6–10] are proposed, which have the limitations that group members are not mutually authenticated, dynamic group membership is not supported, or the cost of key establishment is substantial. Another common limitation of these schemes is that group members must exchange public information to perform key establishment before they start communicating. That is, a public server is needed to keep the public information of all members for user's queries. In some cases, however, it is difficult to maintain a public information center and make it always available to all users. Wong[11] proposed a hierarchy of keys for secure group communications. With the

Wen-Her Yang and Shiuh-Pyng Shieh teach in the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan.

**Correspondence to: Dr. Shiuh-Pyng Shieh, Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30018.*
E-mail: ssp@csie.nctu.edu.tw, URL://www.dsns.csie.nctu.edu.tw/ssp.

Contract/grant sponsor: Lee and MTI Center for Internetworking Research.

Contract/grant sponsor: National Science Council.

Copyright © 2001 John Wiley & Sons, Ltd.

hierarchy of keys, they can reduce the number of renewing key messages. However, their renewing key operation is very complicated; a dedicated server is still needed; group members have to perform many decryption operations when members join or leave. Perrig[12] proposed some collaborative key management protocols. In the protocols, the requirement of a dedicated server for key establishment is eliminated, but the exchange of members' public information is still needed.

***The proposed protocol has the properties of perfect** forward **and** backward privacy.*

This paper proposes a secure key agreement protocol for group communications, where the common group key is collaboratively established by all participants. With an efficient key establishment strategy, the cost of renewing a key operation is significantly reduced when group membership changes. The concept of ID-based schemes[13,14] is used in the proposed protocol for mutual authentication and key establishment, hence key agreement can be efficiently achieved without the aid of a trusted third party or the exchange of members' public information. There is no need for a group member to act as a central controller, such that the overhead of key agreement can be balanced among group members. The proposed protocol has the properties of perfect *forward* and *backward privacy*. Forward privacy ensures that a new joining group member cannot get any past group messages, and backward ensures that a leaving group member cannot get any subsequent group messages. Furthermore, we also address the scalability issue to accommodate our protocol to multicast networks.

This paper is organized as follows. An efficient key agreement protocol for secure group communications is proposed in the next section. Subsequently, the third section describes implementation issues of the new key agreement protocol, and the fourth section gives the protocol analysis. The scalability issue of proposed protocol in multicast networks is addressed in the fifth section. Finally, a conclusion is given in the last section.

Key Agreement Protocol

Generally, a group communication session is initiated by a group creator, and then other members participate individually. Members may leave or join while the session is in progress. According to the action of real group sessions, the new key agreement protocol is divided into four phases: key initiation, group creation, member join, and member departure. The initial phase is performed at the key information center to set up the system. The other three phases are executed among group members to perform mutual authentication and key establishment. Below we describe the four phases of our key agreement protocol.

—Key Initiation Phase—

In the new key agreement protocol, a key information center is set up, which is responsible for generating public and secret information for newly registered users. The steps of system setup are described in Procedure 1. Once the secure group system is set up, the key information center is not needed except when new users request to register. When a new user requests to register, he sends the center his ID. Upon receipt of the user ID, the center repeats steps 4 through 6 in Procedure 1.

The key information center performs the following steps to set up the secure system.

1. Generate two large prime numbers p and q, and let $n = p \cdot q$.
2. Obtain the center's secret information d from the following computation.

$$3 \cdot d(\mathrm{mod}(p-1)\cdot(q-1)) = 1. \quad (1)$$

3. Find an integer g which is a primitive element in both GF(p) and GF(q).
4. Choose a one-way function h to compute the extended identity (EID_i) of i as follows:

$$EID_i \equiv h(1D_i)(\mathrm{mod}2^N), \quad (2)$$

where ID_i is the identity of used i and N denotes the bit length of EID.

5. After computing EID_i, calculate the user secret information S_i as:

$$S_i \equiv EID_i^d(\mathrm{mod}n). \quad (3)$$

Copyright © 2001 John Wiley & Sons, Ltd.

From the relations above, the following equation would be obtained.

$$EID_i \equiv S_i^3 (\mathrm{mod} n) \quad (4)$$

6. Send $(n, g, h(x), S_i)$ back to user i over a secure channel. Upon receipt of the information, user i must keep S_i secret and store the public information $(n, g, h(x))$.

Procedure 1. The steps of system setup

—Group Creation Phase—

Before holding a group session, a group creator has to prepare the member list that contains the information of the members who will attend the group session initially. Note that the group creator does not act as a group controller for issuing the common group key to members, but only for initiating the group. After preparing the member list, the group creator sends the invitation message to all the initial members and then waits for them to join. Upon receipt of the reply messages of members, the group creator announces that the group session is open by multicasting an initiation message. Then, all the joined group members follow the key agreement protocol to establish a common group key.

The key agreement of the proposed protocol is based on a binary key tree hierarchy, which is depicted in Figure 1. In the key tree, each subtree represents a subgroup and each node contains the key shared by the members in the subtree of this node. Consequently, the key of root node denotes the common group key shared by all members. The key tree is constructed from the leaves up to the root. In leaf nodes, the key values are randomly chosen by group members and treated as secret information. The key of a non-leaf node is established by two designated negotiators through the key agreement procedure, which will be described later. The two designated negotiators are randomly chosen out of the left and right subtrees of the non-leaf node respectively. They represent the left and right subgroups to negotiate a common key for this non-leaf node. After the key agreement procedure is completed, the two negotiators should inform other members of the negotiated key value. In each sub-layer of the key tree, the key agreement procedure is repeated to construct the whole key tree hierarchy.

1. If member i wishes to agree a common key with member j, be calculates the following two integers:

$$X_i \equiv g^{3 \cdot f(r_i, C_i)} (\mathrm{mod} n) \quad (5)$$

$$y_i \equiv S_i \cdot g^{2 \cdot f(r_i, C_i)} (\mathrm{mod} n) \quad (6)$$

   Here $f(x, y)$ is a one-way hash function and r_i is a random number chosen by member i. The value of C_i is determined by the role that member i plays. If member i represents a subgroup to perform the key agreement procedure, then C_i is the shared key of the subgroup. Otherwise C_i is the secret information k_i randomly chosen by member i.
2. Member i sends these two integers X_i and Y_i together with ID_i to member j.
3. Member j calculates $EID_i = h(ID_i)$ and checks whether the following equation holds:

$$EID_i = Y_i^3 / X_i^2 \quad (7)$$

4. If the equation holds, member j generates a random number r_j and calculates the common key K_{ji} as follows:

$$K_{ji} = X_i^{f(r_j, C_i)} = g^{3 \cdot f(r_i, C_i) \cdot f(r_j, C_j)} (\mathrm{mod} n) \quad (8)$$

5. Member j then calculates the following three integers:

$$X_j \equiv g^{3 \cdot f(r_j, C_j)} (\mathrm{mod} n) \quad (9)$$

$$Y_j \equiv S_j \cdot g^{2 \cdot f(r_j, C_j)} (\mathrm{mod} n) \quad (10)$$

$$Z_j = \{X_j\} K_{ji} \quad (11)$$

   Here Z_j is the result of encrypting X_j with the common key K_{ji} under symmetric cryptographic algorithms, and is used for member i to verify the correctness of the common key.
6. Member j sends these three integers X_j, Y_j and Z_j along with ID_i to member i.
7. Member i calculates $EID_j = h(ID_j)$ and checks if the following equation holds:

$$EID_j = Y_j^3 / X_j^2 \quad (12)$$

8. If it is true, member i calculates the common key K_{ij} as follows:

$$K_{ij} = X_j^{f(r_i, C_i)} = g^{3 \cdot f(r_i, C_i) f(r_i, C_j)} (\mathrm{mod} n) \quad (13)$$

Copyright © 2001 John Wiley & Sons, Ltd.

9. Member i verifies the correctness of K_{ij} by decrypting Z_j.
10. If member i and j represent subgroups to perform the key agreement procedure, they will inform other members of subgroups the common key respectively. For security's sake, the informing message can be encrypted with the shared key of subgroups.

Procedure 2. The execution steps of key agreement

The key agreement procedure, which is described in Procedure 2, only needs two messages to complete mutual authentication and key establishment. After repeating the key agreement procedure in each sub-layer of the key tree, all group members share the root key as a common group key and know only the keys on the path from their leaf nodes up to the root.

—Member Join Phase—

During a group session, it is possible that a new member requests to join the group occasionally. The new member chooses a joined group member as his *partner*, who locates at the leaf node nearest the root and is responsible for his joining as well as departure. The new member sends his partner the join-request message. If the new member is permitted to join, the partner performs the key agreement procedure with the new member. For providing perfect backward privacy, the joining member cannot get any old keys in the key tree.

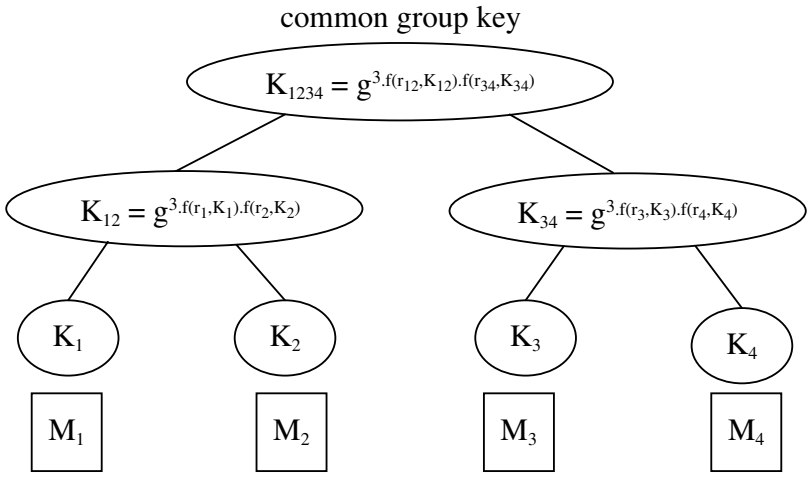

Figure 1. The group key tree hierarchy

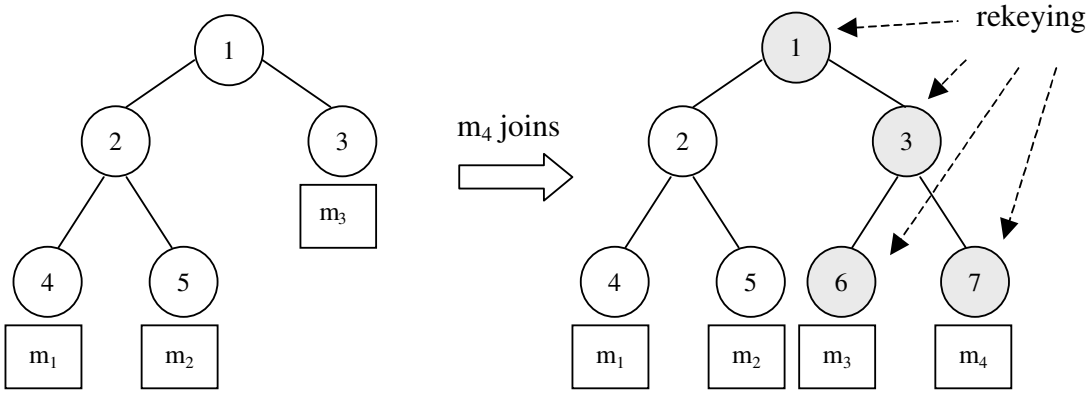

Figure 2. Key tree after the new member m_4 joining

Copyright © 2001 John Wiley & Sons, Ltd.

Hence all the keys from the leaf node of the new member up to the root need to be changed. As an example in Figure 2, the new member m_4 joins at node 7, and member m_3 moves one level down to node 6 for accommodating m_4. The key values of nodes 1, 3, 6 and 7 must be renewed. Here, two issues are not addressed that how the new member finds the partner who is at the leaf node nearest the root, and how the keys are renewed. We will discuss the two issues in the next section.

—Member Departure Phase—

Similarly, it is possible that some group members may want to leave during group sessions, either voluntarily or compulsorily. We should guarantee that the leaving member cannot acquire any information about the subsequent communication contents. That is, all the keys that the leaving number knows must be renewed to provide perfect forward privacy. For example, Figure 3 shows that member m_1 leaves the secure group and member m_2 moves one level up to node 2. The key values of nodes 1 and 2 need to be changed. In next section, we will describe the renewing key operation.

Implementation Issues

In the proposed key agreement protocol, all the group members collaboratively construct the key tree to establish the common group key. In order to achieve this goal, all participants need to know the key structure of subtrees which they belong to. First, the group creator is able to establish an initial key tree structure in the group creation phase, since he has the list of initial group members. He then multicasts the group initiation message, which contains the key tree structure and related initiation information, to all group members. Upon receipt of the initiation message, each member keeps the key structure and performs the mutual authentication procedure to construct the key tree hierarchy collaboratively. If a member joins or leaves a group, group members can adjust the key structure themselves according to the joining and leaving messages. Therefore, no extra message is needed to maintain the consistency of the key structure.

As mentioned earlier, the key value of a non-leaf node is established by two designated negotiators, randomly chosen from left and right subtrees respectively. The problem is how to randomly choose the negotiators. We can simply designate the left most member in the left subtree and the right most member in the right subtree as the default negotiators, or let the group creator assigns the two negotiators for each non-leaf node dynamically. The members having more computation power and better network connectivity can be assigned to non-leaf nodes, so that the key agreement protocol gains better performance in constructing the key tree hierarchy. An alternative

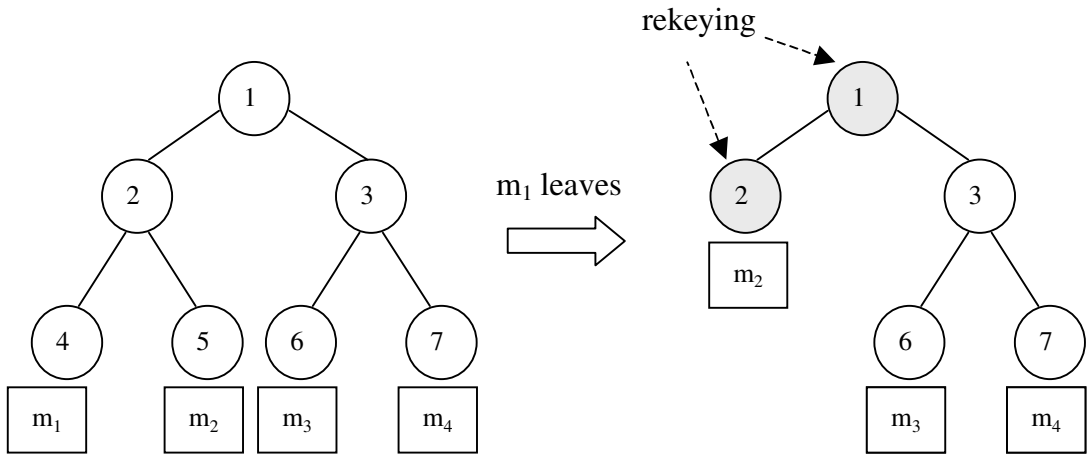

Figure 3. Key tree after leaving the member m_1

Copyright © 2001 John Wiley & Sons, Ltd.

to this issue is not to choose, but let all participants compete for the two negotiators. It means that every member is a negotiator candidate, and takes a random delay to send the authentication messages for the subtrees he belongs to. When a member receives the authentication message sent from the member in the same subtree, he will not need to send the authentication message for the current subtree. It makes load balance in constructing the key tree, since every member has the chance to perform the key establishment operation.

In the member join phase, a new member should find a joined member as his partner before joining the secure group. In the previous section, we assume that the partner should locate at the leaf node nearest the root. This assumption is for keeping the key tree balance for better performance on key establishment. To realize this assumption, the new member can send a join request message to any joined member, when he intends to join the secure group. If the new member is allowed to join, the joined member replies to the key tree structure. Upon receipt of the key tree structure, the new member can choose the appropriate partner to perform the key agreement procedure. In fact, even if the new member does not choose the member who locates at the leaf node nearest the root as his partner, the key agreement protocol still works.

When a member joins or leaves, it represents the insertion or removal of a leaf node in the group tree. Consequently, the group keys of his siblings and ancestors have to be renewed. For leaf nodes, the renewing operation is trivial, since the key values are randomly selected by group members. As to non-leaf nodes, there are two ways to accomplish the renewing key operation. The first is to perform the key agreement procedure to re-negotiate the key values for all the non-leaf nodes. The second way is to randomly choose a new key value and encrypt it with the old key value, then send it to all members. Hence only permitted members can get the new key value. This can be achieved by the partner of the joining/leaving member. If the partner has left the group, a group member in the same subtree is randomly chosen to execute the renewing key operation. For example, in Figure 2, the key values of nodes 6 and 7 are randomly chosen by members m_3 and m_4 respectively. Member m_3 regenerates the key values of nodes 1 and 3, then sends the renewing messages to other group members in the same subtrees. The second way is more efficient than the first but loses the property that shared group keys are cooperatively established by group members.

Exponential computation is considered to be very time-consuming. In order to have better performance, the number of exponential computations for the key agreement procedure may be reduced from five to two with some implementation methods. In equations (5) and (6), we can first compute $g^{f(r_i,C_i)}$, then calculate X_i and Y_i as follows:

$$X_i \equiv g^{f(r_i,C_i)} \cdot g^{f(r_i,C_i)} \cdot g^{f(r_i,C_i)} (\mathrm{mod} n) \quad (14)$$

$$Y_i \equiv S_i \cdot g^{f(r_i,C_i)} \cdot g^{f(r_i,C_i)} (\mathrm{mod} n) \quad (15)$$

No exponential computation but multiplication is needed in these two equations. The verification in equation (7) can also be accomplished without exponential computation in the same way. Therefore, only two exponential computations (one for $g^{f(r_i,C_i)}$ and one for the common key $(X_j)^{f(r_i,C_i)}$) are needed. This makes our key agreement protocol more efficient in establishing the common group key.

In the proposed key agreement protocol, the idea of ID-based schemes is used for mutual authentication and key establishment.

Protocol Analysis

In the proposed key agreement protocol, the idea of ID-based schemes is used for mutual authentication and key establishment. The key agreement procedure has the features that authentication can be efficiently achieved without the aid of a trusted third party or a public information center, and the load of key agreement is balanced and distributed among all group members. It does not have the conspiracy problem existing in the Tsujii's[15] scheme because its security relies on the difficulty of computing the discrete logarithm problem. If a forger wants to persuade member i to join the group, he must find two integers X and Y satisfying the verification of equation (7). The use of low public exponents in this equation does not lower the difficulty to crack (Y,X). Although the forger can get a

Copyright © 2001 John Wiley & Sons, Ltd.

pair of integers (Y^3, X^2) to make the equation hold, the pair (Y, X) is unattainable because computing (Y, X) from (Y^3, X^2) is a discrete logarithm problem.

Hastad[16] proposed an attack on RSA with low exponents in a public key system. The attack will not succeed in our key agreement procedure, since the same modulus n is used for all members. An outsider, even a departed member, cannot crack the common group key, since he has no idea of all $g^{f(r_i, C_i)}$ to compute the group key. Although a departed member has the old group key, he is unable to get the new $g^{f(r_i', C_i')}$, which is regenerated after the member has left, to derive the new group key. Suppose an intruder has intercepted a pair (Y_A, X_A) of the key agreement procedure between members A and B, he then chooses a random number R and computes a new pair $(Y_A' = Y_A R^2, X_A' = X_A R^3)$. If the intruder sends the new pair (Y_A', X_A') to member B, the verification of equation (7) will success because

$$Y_A'^3 / X_A'^2 = Y_A^3 R^6 / X_A^2 R^6 = Y_A^3 / X_A^2 = EID_A. \quad (16)$$

This implies that member B may derive the wrong group key $K_{AB}' = (X_A')^{f(r_B, C_B)}$. However, this attack can be detected in step 9 of Procedure 2. When member B sends (X_B, Y_B, Z_B) back to member A, member A will check the integrity of group key $K_{AB} = (X_B)^{f(r_A, C_A)}$ by decrypting Z_B. Since K_{AB} is not equal to K_{AB}', the decryption will fail and the authentication message (X_B, Y_B, Z_B) is regarded as a forgery. Consequently, the *man-in-the-middle* attack will fail.

Each member has to perform the key agreement procedure before joining the secure group. Hence only authenticated members can get the common group key. A feature of the proposed key agreement protocol is that each member only knows the keys on the path from his leaf nodes up to the root in the key tree. This feature makes the renewing key operation more efficient since not all the keys in the tree hierarchy need to be renewed. With an efficient renewing key operation, new joining members cannot get any past group messages and leaving members cannot acquire any subsequent communication. This ensures that the proposed key agreement protocol provides perfect forward and backward privacy.

	Our protocol	A-GDH.2 (reference 9)	SA-GDH.2 (reference 9)	NAGKA (reference 12)	AGKA (reference 12)
Execution rounds	$d = \log N$	N	N	d	d
Exponential computations per member (min, max)	2, 2d	1, N	N, N	$d + 1$, 2d	$d + 1$, 2d
Messages sent per member (min, max)	1, d	1, 1	1, 1	1, d	1, d
Messages received per member (min, max)	d, d	2, 2	2, 2	d, d	d, d
Total messages	$2(N - 1)$	N	N	$2(N - 1)$	$2(N - 1)$
Mutual authentication of members	Yes	No	Yes	No	Yes
Need of a group controller	No	Yes	Yes	No	No
Need of public information exchange	No	Yes	Yes	Yes	Yes

Table 1. Comparisons of proposed protocol with other schemes

Copyright © 2001 John Wiley & Sons, Ltd.

We evaluate the performance of the proposed protocol on both communication and computation complexities. The measures of communication complexity are the number of messages transmitted for key establishment. The computation complexity is measured by the number of exponential operations, because they are considered to be more time-consuming than other arithmetic operations. Since the key tree hierarchy is used in our protocol, only log$N$ rounds of key agreement operations are needed to establish the common group key among N members. Moreover, mutual authentication is achieved with only two message exchanges and two exponential computations for per key agreement operation. Table 1 lists the comparisons of our protocol with other schemes. We assume herein that the number of group members is N.

Consideration of Multicast Networks

Since ID-based authentication is adopted in the proposed protocol, group members are required to register at the same key information center. For local networks, this is reasonable and the proposed protocol works securely and efficiently. In an open environment such as the Internet, however, it is difficult to set up a global key information center for all members. This scalability issue may be released by considering the topology of multicast networks into the proposed protocol. That is, the key agreement protocol is used to establish local group keys to provide the security of local domains; the group messages flowing across domains can be protected by introducing a security mechanism into the multicast routing protocols. To deliver multicast traffic to all receivers in multicast networks, multicast distribution trees are used to describe the path that the multicast traffic takes through the networks.

Multicast distribution trees are divided into two categories: *source trees* and *shared trees*. In source trees, the source of the multicast traffic is the root to form a spanning tree through the network to the receivers. On the other hand, shared trees use a single common root at some chosen

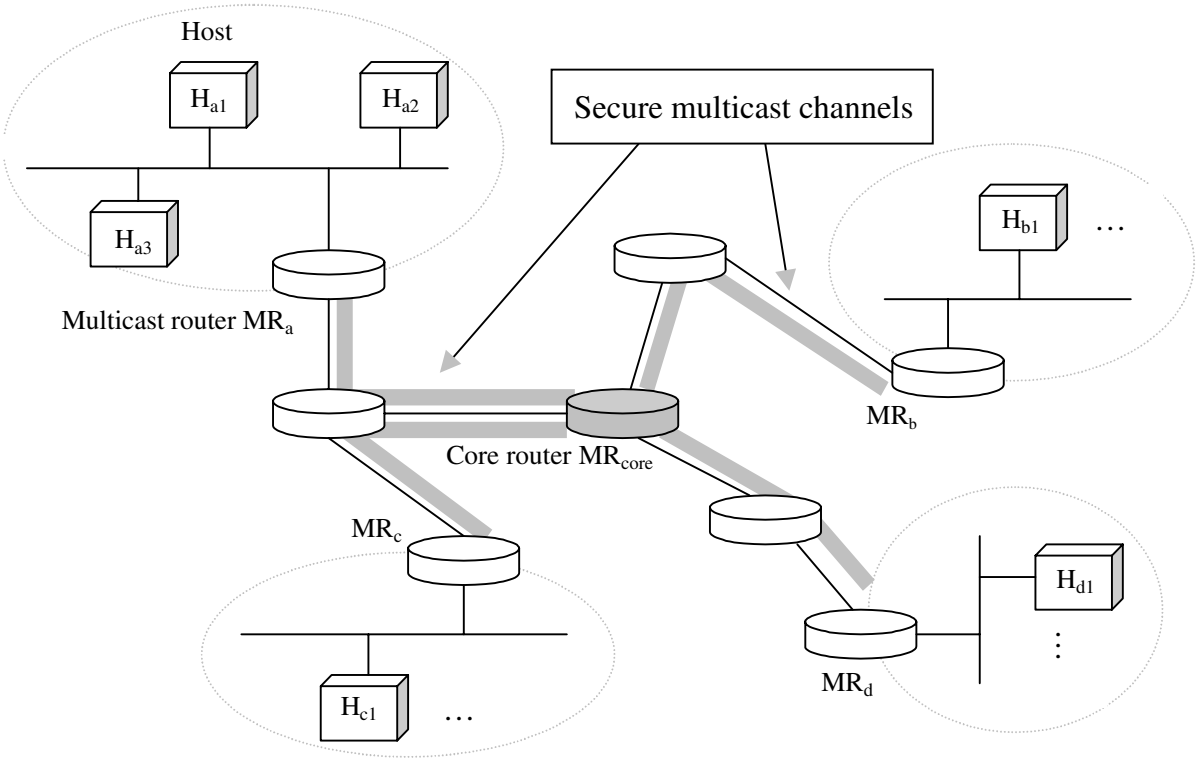

Figure 4. The secure multicase channels

Copyright © 2001 John Wiley & Sons, Ltd.

point in the network to form a shared tree for each multicast group. Depending on the multicast routing protocols (CBT[17] or PIM-SM[18] the common root is called a *core* or *rendezvous point*. In multicast routing protocols with shared trees, a receiver's local multicast router is required to send the core router a request to explicitly join the corresponding shared tree after receiving an IGMP[19] group membership report message. With this property, we may introduce a security mechanism, which is transparent to communication participants, into the multicast routing protocols.

Without loss of generality, the network topology depicted in Figure 4 is used to describe the security mechanism. When a user start the key agreement protocol on a host said H_{a1} to join a secure multicast group, host H_{a1} immediately sends an IGMP group membership report to inform the local designated router MR_a that he wants to join a multicast group. Upon receipt of the IGMP report, MR_a notices that it is the first report to the multicast group, he also performs the key agreement protocol to get the local group key, then generates a signed join-request for the shared distribution tree and sends to the core router MR_{core}. Upon receipt of the join-request, MR_{core} authenticates MR_a by verifying the signature. If successful, MR_{core} will generate a channel encryption key for the multicast group and send back a join-ack along with the channel key encrypted with the public key of MR_a. The channel key is used to decrypt the group messages subsequently received from MR_{core}. Consequently, the core router MR_{core} establishes both the shared distribution tree and the secure multicast channels at the same time. The secure multicast channels shown in Figure 4 protect the group messages flowing on the public networks. After then, if MR_a receives group messages from local domain, he will decrypt the messages with the local group key and send the messages re-encrypted with the channel key to the core router MR_{core}. Here the message re-encryption overhead can be reduced by introducing one more message encryption key. When a group member wants to send messages, he generates a random key to encrypt the group messages and the random key is encrypted with the local group key before multicasting. The local router will receive the encrypted messages along with the random key protected by the local group key, such that he can only re-encrypt the random key with the channel key and forwards these information to other multicast routers. Therefore, the extra message encryption key eliminates the overhead of re-encrypting the full messages, and the privacy of group messages in both local and public networks is guaranteed.

Conclusions

This paper proposes an efficient key agreement protocol for secure group communications, in which the common group key is collaboratively established by all members. With the key tree hierarchy, the establishment of a common group key requires only log$N$ rounds of key agreement operations among N members. Unlike other schemes, the proposed protocol does not need a dedicated server or group controller for key distribution. All participants in the secure group are mutually authenticated by performing the key agreement procedure. There is no need to exchange the public information of group members for the key agreement procedure. This feature makes the proposed protocol more suitable for many group communication environments. The dynamic membership issues are also considered in our protocol. For security, the group keys have to be renewed when members join and leave. The feature that each member knows only the keys on the path from his leaf nodes up to the root in the key tree makes the renewing key operation more efficient. With an efficient renewing key operation, the properties of perfect forward and backward privacy are ensured. The proposed protocol also can be applied to multicast networks with accommodation to multicast routing protocols.

Acknowledgements

This work is supported in part by Lee and MTI Center for Internetworking Research (Global Crossing), Ministry of Education, and National Science Council.

References

1. Harney H, Muckenhirn C. Group key management protocol (GKMP) specification/architecture. *Technical Report RFC-2093 and RFC-2094*, IETF, July 1997.

Copyright © 2001 John Wiley & Sons, Ltd.

2. Mittra S, Iolus. A framework for scalable secure multicasting. In *ACM SIGCOMM*, Sept. 1997.
3. Chang I, Engel R, Kandlur D, Pendarakis D, Saha D. Key management for secure internet multicast using Boolean function minimization techniques. *INFOCOM 1998*, Sept. 1998.
4. McGrew DA, Sherman AT. Key establishment in large dynamic groups using one-way function trees. http://www.cs.umbc.edu/~ sherman/Papers/itse.ps, May 1998.
5. Wallner DM, Harder E, Agee RC. Key management for multicast: issues and architectures, *Technical Report IETF*, draft-wallner-key-arch-01.txt, Sept. 1998.
6. Burmester M, Desmedt Y. A secure and efficient conference key distribution system. *EUROCRYPTO94*, pp. 275–286, 1994.
7. Ballardie A. Scalable multicast key distribution. *Technical Report RFC-1949*, IETF May 1996.
8. Oppliger R, Albanese A. Participant registration, validation, and key distribution for large-scale conferencing systems. *IEEE Communications Magazine* June 1997; **35**: 130–134.
9. Ateniese G, Steiner M, Tsudik G. Authenticated group key agreement and friends. In *5th ACM Conference on Computer and Communications Security*, pp. 17–26, Nov. 1998.
10. Sun HM, Shieh SP. Secure broadcasting in large networks. *Computer Communications* March 1998; **21**, No. 3, 279–283.
11. Wong CK, Gouda M, Lam SS. Secure group communications using key graphs. *Proceedings of ACM SIGCOMM'98*, pp. 68–79, Sept. 1998.
12. Perrig A. Efficient collaborative key management protocols for secure autonomous group communication. *International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC'99)*, 1999.
13. Shamir A. Identity-based cryptosystems and signature schemes. *Proceedings CRYPTO'84*, pp. 47–53, Springer: Berlin, 1985.
14. Shieh SP, Yang W, Sun HM. An authentication protocol without trusted third party. *IEEE Communications Letters* May 1997; **1**: No. 3.
15. Tsujii S, Itho T, Kurosawa K. ID-based cryptosystem using discrete logarithm problem. *Electronic Letters* Nov. 1987; **23**: 1318–1320.
16. Hastad J. On using RSA with low exponent in a public key network. Lecture Notes in Computer Science: *Advances in Cryptology-CRYPTO'85 Proceedings*. Springer-Verlag: New York; 403–408.
17. Ballardie AJ. Core Based Trees (CBT) multicast: architectural overview. Internet draft draft ietf-idmr-cbt-arch-02.txt, June 20, 1995.
18. Deering S, Estrin D, Farinacci D, Jacobson V, Liu C, Wei L, Sharma P, Helmy A. 'Protocol-independent multicast-sparse mode (PIM-SM): protocol specification. Internet draft draft-ietf-idmr-PIM-SM-spec-02.ps, Sep. 7, 1995.
19. Deering S. Host extensions for IP multicasting. Internet RFC 1112, August 1989.
20. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signature and public-key cryptosystem. *Communication of the ACM* 1978; **21**, No. 2, 120–126. ■

If you wish to order reprints for this or any other articles in the *International Journal of Network Management*, please see the Special Reprint instructions inside the front cover.

Copyright © 2001 John Wiley & Sons, Ltd.