INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2001; **11**:129–136

A secure multicast protocol for the internet's multicast backbone

By Wen-Her Yang, Kai-Wei Fan and Shiuh-Pyng Shieh[*]

In this paper we propose a secure and efficient multicast protocol where the key management is distributed to local groups. The proposed protocol takes advantage of MBone topology to maintain scalability and efficiency at the same time. Copyright © 2001 John Wiley & Sons, Ltd.

Introduction

In recent years, the Internet has seen an increase in the number of new applications that rely on multicast transmission. Multicast is a technology that conserves bandwidth in point-to-multipoint transmission. With the development of Mbone technology, an Internet-wide virtual network for point-to-multipoint transmission, group communication becomes possible NetMeeting, NetShow and many other applications for group communications are available on the Internet. The Internet's Multicast Backbone (Mbone) is the small subset of Internet routers and hosts that are interconnected and are capable of forwarding IP multicast traffic. The MBone constructs a virtual network that is divided into sub-networks called islands. The islands are connected by multicast-capable routers via virtual point-to-point links called 'tunnels'. The tunnels enable multicast traffic to pass through the non-multicast-capable parts of the Internet. Multicast packets sent to a local island are captured by a local Multicast router (MRouter), then the MRouter encapsulates these packets in IP-over-IP format and unicasts to other MRouters in remote islands via tunnels. The remote MRouters then strip the encapsulated packets and multicast to their local islands.

Since the Internet is an open environment, multicast messages may be eavesdropped easily. In order to prevent intruders from cracking group communications, a secure multicast environment must be provided. Confidential data have to be encrypted before transmission and only legal group members can acquire communication contents. The basic requirement is that all group members need to know the common group key that is used to encrypt communication data. Then the main problem is how to distribute secret information or a common key to the members distributed across the Internet and a secure key-establishment process is required. Key-establishment schemes can be classified into two categories: one is key agreement and the other is key distribution. Key agreement allows all members to determine the common key securely and collaboratively. The Diffie–Hellman key exchange[1] is an example of key agreement techniques for two parties. Key distribution is simpler, it only needs someone to select the common key and securely distribute it to others. There are two kinds of key distribution: centralized and decentralized. Centralized schemes usually

Wen-Her Yang and Kai-Wei Fan Teach in the Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan.

Shiuh-Pyng Shieh received the M.S. and Ph.D degrees in electrical and computer engineering from the University of Maryland in 1986 and 1991 respectively. He is currently the Director of the Computer and Network Center; a Professor with The Department of Computer Science and Engineering, National Chiao Tung University, Taiwan; and Vice President of The Chinese Cryptology and Information Security Association. His research interests include internetworking, distributed operating systems, and network security.

[*] *Correspondence to: Shiuh-Pyng Shieh, Director and Professor, Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30010.*
Email: ssp@csie.nctu.edu.tw
http://www.dsns.csie.nctu.edu.tw/ssp

require a trusted third party to be a key distribution center (KDC), while decentralized schemes do not need it and are usually public key based.

Many related schemes[2–5] for establishing secure group communications have been proposed. However, these schemes either are designed for fixed tree structures or high-cost key-renewing operations. In 1993, Rolf Oppliger[6,7] proposed the DiRK (Distributed registration and key distribution) scheme in which participant registration and key distribution is performed in a decentralized way. However, the key-renewing process of DiRK is not efficient because all group members are involved. Another key distribution scheme is Scalable Multicast Key Distribution (SMKD) defined in RFC-1949,[8] which provides a scalable solution to multicast key distribution. Since all group members are involved, the scheme incurs the high cost of a key-renewing operation. Thomas Hardjono and Brad Cain[9] also proposed a group key management scheme for N-to-N Multicast. Though they stated that their proposed scheme is practical on Mbone and has better scalability, the cost of the key-renewing operation is still high.[10] Wong *et al*.[10] proposed a hierarchy of keys for secure group communications. With the key hierarchy, the key-renewing messages can be reduced. However, the key-renewing process is very complicated and a dedicated server is still needed. Group members have to perform many decryption operations when members join or leave. Sun and Shieh[11] proposed another scheme in which users only need to store one subgroup key. Since group hierarchy and members' relations are fixed upon the creation of the group, the scheme is only suitable for a fixed tree structure, such as a shared delivery tree where group membership and the topology of the group will not change. Therefore, the scheme cannot be employed on Mbone directly.

In this paper, a scalable and secure multicast protocol that is suitable for MBone is proposed. The proposed protocol takes advantage of the network topology of MBone to make it more efficient on key management and distribution. In our protocol, group members are divided into local subgroups. Each subgroup belongs to an island that is physically a subnetwork on MBone. When users join or leave a group, the key-renewing process will be confined to a local group. That is, only members in the same subgroup need to renew the subgroup key, where joining or leaving operations occur. The characteristic of MBone networks makes our protocol more efficient and scalable, since the key-renewing process can be localized and simplified. Our protocol contains two modes that can easily adapt to different group communication environments to get better performance. This paper is organized as follows. In the next section a scalable secure multicast protocol is proposed. The third section discusses the operation modes of the proposed protocol to adapt to different group communication behaviors. Then, security analysis and performance evaluation will be provided in the fourth section. Finally, the fifth section gives a conclusion.

The characteristic of MBone topology is considered as an important factor in designing an efficient protocol that can satisfy data confidentiality as well as lower computation and communication overhead.

Scalable Secure Multicast Protocol

In this section a scalable and secure multicast protocol suitable for MBone is proposed. The characteristic of MBone topology is considered as an important factor in designing an efficient protocol that can satisfy data confidentiality as well as lower computation and communication overhead. The proposed protocol employs a distributed method to achieve key management and distribution without the participation of a key distribution center. In our protocol, the key-renewing process is confined to local islands when users join and leave. This feature results in the low cost of the key-renewing operation. In general, a secure multicast protocol is divided into four phases; key management, group creation, member join and member departure. We will present the proposed protocol in the four phases in detail. For convenience, the notation listed in Table 1 is used to describe the proposed protocol.

—Key Management—

In MBone networks the multicast topology can be divided into *backbone* and *leaf islands*.

Notation	Description
m_{ij}	Member j in subgroup G_i
G_i	Subgroup(subnetwork) i, $G_i = \{m_{i1}, m_{i2}, m_{i3}, \ldots\}$
G	Multicast group, $G = \{G_1, G_2, G_3, \ldots\}$
MR_i	MRouter in subgroup G_i
S	Secret information
K_{gi}	Subgroup key of G_i
K_E	Message encryption key
$\{M\}K_E$	Encryption of message M with K_E

Table 1. Notation definitions

All leaf islands are interconnected by MRouters. While transmitting multicast packets to remote islands, MRouters deliver packets through the tunnels between adjacent MRouters. Since MBone is divided into islands physically, an island is considered as a subgroup in the multicast group. In our protocol, only group members can derive the group common key, and only subgroup members know the secret key of the subgroup to which they belong. Although MRouters are responsible for routing multicast packets, it is more desirable that MRouters have no knowledge of the communication contents. Figure 1 is an example of a multicast group topology on MBone. We assume that there are m subgroups $(G_1, G_2, \ldots, G_m)$ in the multicast group, and each MRouter MR_i is responsible for a subgroup G_i.

Key generation— To establish a secure multicast session, several group parameters should be set up. They are described as follows:

1. The group creator (session holder) computes a large integer $n = p \cdot q$, and $\varphi(n) = (p-1)(q-1)$. Here p and q are large prime numbers that satisfy the RSA assumption[12], and n should be published.
2. The group creator chooses a random number as the secret information S, which should be kept privately, and generates a shared key K_{MR} distributed to all joined MRouters securely.
3. For each subgroup G_i, a pair of e_i and d_i is generated such that $e_i \cdot d_i \bmod \varphi(n) = 1$. Here e_i and d_i are only known by the group creator as well as MRouter MR_i.
4. Each subgroup member in island G_i obtains the subgroup key $K_{gi} = S^{e_i} \bmod n$ securely from the group creator.

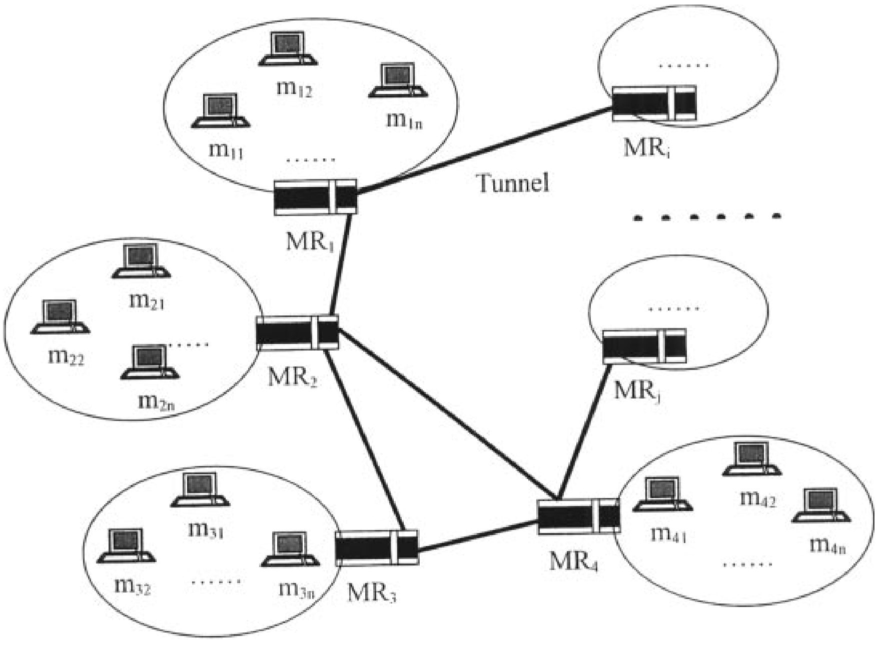

Figure 1. The multicast topology

Note that MRouter MR_i only knows n and d_i, he has no idea about the secret information S and $\varphi(n)$. Consequently, MRouter MR_i cannot derive the subgroup key K_{gi} to acquire the message contents.

Message encryption and transmission— After the group parameters have been set up, users can start the secure multicast session. Suppose a member m_{i1} in subgroup G_i wants to securely multicast a message M; the following steps are performed:

1. The sender chooses a random number r and computes $K_E = K_{g_i}^r = S^{e_i r} \bmod n$ as the message-encryption key.
2. After generating the encryption key K_E, the sender encrypts the message M with K_E and sends the message $\{r, \{M\}K_E\}$ as a multicast packet.
3. Upon receipt of the multicast packet $\{r, \{M\} K_E\}$, MRouter MR_i first computes $\{e_i \cdot r\}$ and encrypts it with K_{MR}, then routes the modified packet $\{\{e_i \cdot r\}K_{MR}, \{M\}K_E\}$ to other MRouters in remote islands.
4. As to each remote MRouter MR_j received the multicast packet $\{\{e_i \cdot r\}K_{MR}, \{M\}K_E\}$, he will simply route the packet to neighboring MRouters, and simultaneously decrypt $\{e_i \cdot r\}K_{MR}$ to modify the packet as $\{d_j \cdot e_i \cdot r, \{M\}K_E\}$, then multicast the modified packet to his local island.

Message decryption— When members receive the multicast packet from m_{i1}, they must derive the message-encryption key K_E to decrypt it. The decryption procedure is as follows:

1. Members in the same subgroup G_i as the sender m_{i1} will receive the packet $\{r, \{M\}K_E\}$. Since they share the same subgroup key K_{gi} with m_{i1}, they can derive the message encryption key $K_E = K_{g_i}^r = S^{e_i r} \bmod n$ to decrypt $\{M\}K_E$ directly.
2. Members in other subgroup G_j will receive the packet $\{d_j \cdot e_i \cdot r, \{M\}K_E\}$ from their local MRouter MR_j. Similarly, they can derive $K_E = K_{g_j}^{d_j e_i r} = S^{e_i d_j e_i r} = S^{e_i r} \bmod n$ to decrypt the packet.

Figure 2 is an example of message multicasting in the proposed scheme. The sender m_{l1} in subgroup G_l multicasts a message M and the encryption key is $K_{g_1}^r \bmod n$. Members in subgroup G_l can derive the encryption key from r and the subgroup key K_{gl} to decrypt $\{M\}K_E$. The MRouter MR_i modifies the received packet to $\{\{e_l \cdot r\}K_{MR}, \{M\}K_E\}$ and routes it to remote islands. Upon receipt of the packet, MR_2

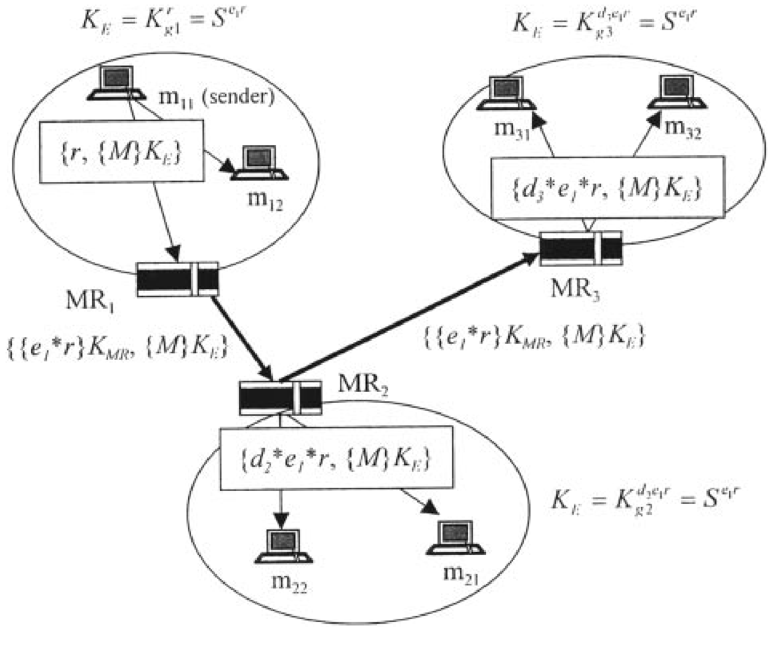
$K_E = K_{g1}^r = S^{e_1 r}$
m₁₁ (sender)
$\{r, \{M\}K_E\}$
m₁₂
MR₁
$\{\{e_1*r\}K_{MR}, \{M\}K_E\}$
$K_E = K_{g3}^{d_3 e_1 r} = S^{e_1 r}$
m₃₁
m₃₂
$\{d_3*e_1*r, \{M\}K_E\}$
MR₃
$\{\{e_1*r\}K_{MR}, \{M\}K_E\}$
MR₂
$\{d_2*e_1*r, \{M\}K_E\}$
$K_E = K_{g2}^{d_2 e_1 r} = S^{e_1 r}$
m₂₂
m₂₁

Figure 2. The process of message multicasting

re-routes it to MR_3. MR_2 and MR_3 then modify the packet to $\{d_2 \cdot e_l \cdot r, \{M\}K_E\}$ and $\{d_3 \cdot e_l \cdot r, \{M\}K_E\}$ respectively, and multicast it to their own islands. The members in those islands can derive the encryption key from their own subgroup key K_{g2} and K_{g3}, and decrypt the packet to obtain M. Note that in the proposed scheme, group members only need know their own subgroup key, hence the key management is confined to local groups.

—Group Creation—

Before group members can start to communicate on MBone, a multicast group must be created. As mentioned in the previous subsection, the group creator (or session holder) first generates the modulo n, $\varphi(n)$, secret information S and the key pair (e_i, d_i) for his local MRouters MR_i. He then selects an unused multicast address and notifies MR_i that he wants to use it as the multicast group address. Here the key pair (e_i, d_i) is delivered to MR_i via a secure channel at the same time. Upon receipt of MR_i's reply, the group creator computes the subgroup key $K_{gi} = S^{e_i} \bmod n$ and announces that the multicast group is created.

—Member Join—

During the group session it is possible that a occasionally user requests to join the group. In order to communicate with others securely, the joining user has to acquire the subgroup key. If the joining user is not the first member in the island, the subgroup key must be renewed to prevent him from acquiring any information about the past communication contents. The joining process is as follows:

1. When a user wants to join the group, he sends the joining request to the local MRouter MR_j, then MR_j forwards the request to the group creator.
2. If the user is allowed to join the group, the group creator generates a new key pair (e_j', d_j') such that $e_j' \cdot d_j' \bmod \varphi(n) = 1$. After computing the new subgroup key $K_{gj}' = S^{e_i'} \bmod n$, the group creator then sends (e_j', d_j') to MR_j and K_{gj}' to the joining user respectively via secure channels.
3. If the joining user is not the first member in the island, MR_j should replace the old key pair (e_j, d_j) with the new one (e_j', d_j'). Besides, the MRouter MR_j will notify other members to renew the subgroup key by multicasting $\{e_j' \cdot d_j\}$ to the local island.
4. Upon receipt of the key-renewing message $\{e_j' \cdot d_j\}$, the joined members in the local island compute the new subgroup key $K_{g_i}^{e_i' d_i} = S^{e_i'} \bmod n$. Then all subgroup members share a new subgroup key K_{gj}'.

Note that the joining process only occurs in the local island, members in other islands are not aware of the joining process and need to do nothing. This property significantly reduces the overhead of the key-renewing operation.

—Member Departure—

When a group member wants to leave the secure multicast group, we should make sure that the leaving member cannot acquire any information about the future communication contents. Thus the subgroup key has to be renewed when members leave. The leaving process is as follows:

1. If member m_{ij} wants to leave the group, he sends the leaving request to the local MRouter MR_j, then MR_j forwards the request to the group creator.
2. Upon receipt of the leaving request, the group creator generates a new key pair (e_j', d_j') such that $e_j' \cdot d_j' \bmod \varphi(n) = 1$, then sends the new key pair to MR_i securely.
3. Upon receipt of the new key pair (e_j', d_i'), MR_j computes $\{e_i' \cdot d_j\}$ and distributes it to all the remaining members as the key-renewing message. There are many ways to distribute secret information to others. Here, we will not discuss the detail of the key distribution procedure. In the worst and simplest case, MR_j can just unicast the secret information to all the remaining members in the local island through secure channels, or we can adapt the key distribution methods proposed in reference.[13] After receiving the key-renewing message $\{e_j' \cdot d_j\}$, the remaining members will compute $K_g' = K_g^{e'd} = S^{e'} \bmod n$ as the new subgroup key.

Like the joining process, the leaving process only occurs in the local island, and the other members in remote islands will not be involved. Since the user joining and leaving processes are both confined to a local group, the cost of the key-renewing operation will be reduced significantly.

Protocol Operation Modes

In the previous section we presented a secure multicast protocol that is scalable and efficient in the key-renewing operation. To ensure forward and backward privacy, the encryption key is newly generated for each message transmission in the proposed protocol. This strategy may incur a little computation overhead in key generation. In fact, it is not necessary to regenerate a new encryption key for each message transmission if the group membership is unchanged. For example, in an extreme case, the group membership is static. All members can always use the same encryption key for all the message transmissions, since it is not necessary to consider the issue of forward and backward privacy. Thus the performance can be improved if we accommodate the proposed protocol to different group behaviors. The basic idea is to reduce the frequency of encryption key generation.

We divide the proposed protocol into two operation modes. One is *static mode* and the other is *dynamic mode*. In static mode, all group members use a common encryption key for message transmissions, such that the cost of encryption key generation is eliminated. When a group is created, it operates in static mode initially. The common encryption key can be generated by the member who submits the first message. Group members who received the first encrypted message $\{r, \{M\}K_E\}$ as mentioned in the previous section, compute the decryption key K_E and store it as well as the value r for future use. When the next message $\{r', \{M\}K_E\}$ is received, the members check if r' is identical to the value r stored. If it is true, they can use the key K_E for the previous message to decrypt this message directly. Similarly, the members can use K_E as the common encryption key for the messages they want to send. Therefore, both senders and receivers need not regenerate the message-encryption keys. Since the encryption key is not renewed, the proposed protocol operates in this mode when the group membership is unchanged. If members join or leave, all the other members notice the change of membership and switch into dynamic mode. In dynamic mode, all members follow the scenario described in the previous section to perform secure group communications. That is, every member will generate a new encryption key for each message transmission. If no member joins or leaves for a period of time, the group switches into static mode again. The protocol will operate in static mode when the membership of a group tends to remain stable. Consequently, the computation overhead of encryption key generation is significantly reduced.

Security Analysis and Performance Evaluation

In this section we will show our protocol is secure and only legal group members can obtain communication contents. First, an outsider, even a departed member, cannot decode both the current subgroup key of an island and the message-encryption key, since the secret information $K_{gj} = S^{e_i} \bmod n$ is securely distributed to all group members. Although a departed member has the old subgroup key, he is unable to know the value S and e_i to derive the current subgroup key $K_{gj} = S^{e_i} \bmod n$. As to the message-encryption key, the departed member cannot compute $K_E = S^{e_i r} \bmod n$ from knowing only the random number r. In the proposed protocol, MRouters are responsible for routing secure multicast packets and renewing subgroup keys, but they do not belong to the secure group and are not allowed to know communication contents. The MRouter MR_i in charge of subgroup G_i only knows the key pair (e_i, d_i) and n. Thus MR_i is unable to derive the subgroup key $K_{gj} = S^{e_i} \bmod n$ without the knowledge of secret information S.

Now, we consider if the subgroup keys or message-encryption keys are compromised by common attacks. The first attack is substitution. An attacker may substitute one or more messages sent by others. In this protocol, the attacker may substitute some parts of the key-renewing messages. We did not describe the detail of distributing key-renewing messages earlier. Nevertheless, the substitution can be easily detected by applying the sender's signature to the key-renewing message. The other attack is replay. An attacker may impersonate the MRouter or the group creator to send key-renewing messages. Since we can apply a signature and a nonce to each key-renewing message, the replayed key-renewing message will be detected by receivers. Thus the replay attack can be easily prevented.

The proposed protocol employs a distributed way to achieve key management and make multicast communications more efficient.

The proposed protocol employs a distributed way to achieve key management and make multicast communications more efficient. Here, we will compare our protocol with other schemes. The comparison is made by our protocol with the schemes proposed by Hardjono and Cain[9] and Wong98 *et al.*[10] We use SMP to represent our secure multicast protocol, and the schemes in Wong *et al.* and Hardjono and Cain are denoted by SGC and GKM respectively. For convenience, the notation in Table 2 is used. The compared result is listed in Table 3.

There are some fields that have two values in the *SMP* and *GKM* schemes. That is because when users join or leave, the joining or leaving operations only happen in one subgroup. The members in other subgroups will not be aware of the operation. The first value is the cost in non-operating subgroups, and the second is the cost in the operating subgroup. We can see that our protocol has better performance than *SGC* and *GKM* schemes in general.

N	Number of group members
N_i	Number of group members in subgroup G_j
M	Number of subgroups
h	The height of the key tree
d	The maximum edges of a node in the key tree

Table 2. The notations

Conclusion

In this paper we proposed a secure multicast protocol, in which a key distribution scheme suitable for MBone networks is employed. The MRouter of each island on MBone aids subgroup members to renew the subgroup key. When joining or leaving events occur, all key-renewing operations are confined to local islands and only local group members have to participate in the

		SMP	*SGC*			*GKM*
			Star	*Tree*	*Complete*	
Total number of keys		M	$N+1$	$d/(d-1)^{*}N$	$2^N - 1$	$M+1$
Number of keys per user		2	2	h	2^{N-1}	2
Computation cost of a joining user		1	1	$h-1$	2^N	2
Computation cost of non-requesting user in joining operation		0,1	1	$d/(d-1)$	1	1,2
Computation cost of non-requesting user in leaving operation		0,1	1	$d/(d-1)$	0	1,2
Computation cost of server	Global	1	2	$2(h-1)$	2^{N+1}	M
in joining operation	Local	0,2				1,2
Computation cost of server	Global	0	$N-1$	$d(h-1)$	0	M
in leaving operation	Local	0, $N_i - 1$				1,$N_i - 1$

Table 3. Comparisons with other schemes

key-renewing process. This property reduces the cost of key-renewing operations significantly, and scalability of the proposed protocol is achieved. In addition, the privacy of confidential data is guaranteed. Only group members can derive the contents of transmitted messages. Outsiders, even the MRouters responsible for routing secure multicast traffic have no idea of communication contents. For better performance the proposed protocol is divided into two operation modes, *static* and *dynamic*. With accommodation for different group behaviors, the computation overhead of encryption key generation is significantly reduced.

References

1. Diffie W, Hellman ME. New directions in cryptography. *IEEE Transaction on Information Theory* November 1976; No. IT-22, 644–654.
2. Ateniese G, Steiner M, Tsudik A. Authenticated group key agreement and friends. In *5th ACM Conference on Computer and Communications Security* November 1998; 17–26.
3. Chang I, Engel R, Kandlur D, Pendarakis D, Saha D. Key management for secure Internet multicast using Boolean function minimization techniques. *INFOCOM 1999*, September 1999.
4. Harney H, Muckenhirn C. Group key management protocol (GKMP) specification/architecture. *Technical Report RFC-2093 and RFC-2094*, IETF, July 1997.
5. Perrig A. Efficient collaborative key management protocols for secure autonomous group communication. *International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, 1999.
6. Oppliger R, Albanese A. Distributed registration and key distribution (DiRK). *Proceedings of the 12th International Conference on Information Security (IFIP SEC '96)*, Island of Samos (Greece), 21–24 May, Chapman & Hall, London, 199–208.
7. Oppliger R, Albanese A. Participant registration, validation, and key distribution for large-scale conferencing systems. *IEEE Communications Magazine* June 1997; **35**.
8. Ballardie A. Scalable multicast key distribution. *Internet RFC 1949*, May (1996).
9. Hardjono T, Cain B. Secure and scalable interdomain group key management for N-to-N multicast. *Proceedings 1998 International Conference on Parallel and Distributed Systems*. 1998; 478–485.
10. Wong Chung Kei, Gouda M, Lam SL. Secure group communications using key graphs. *Proceedings of ACM SIGCOMM'98*, ACM, September 1998.
11. Sun Hung-Min, Shieh Shiuh-Pyng, Secure broadcasting in large networks. *Computer Communications*, March 1998; **21**: 279–283.
12. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signature and public-key cryptosystem. *Communications of the ACM* 1978; **21**:(2), 120–126.
13. Burmester M, Desmedt Y. A secure and efficient conference key distribution system. *Proceedings of EUROCRYPT'94* 1994; 275–286. ■