INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2004; **14**: 167–176 (DOI: 10.1002/nem.515)

Scalable and lightweight key distribution for secure group communications

By Fu-Yuan Lee[*†] *and Shiuhpyng Shieh*

Securing group communications in dynamic and large-scale groups is more complex than securing one-to-one communications due to the inherent scalability issue of group key management. In particular, cost for key establishment and key renewing is usually relevant to the group size and subsequently becomes a performance bottleneck in achieving scalability. To address this problem, this paper proposes a new approach that features decoupling of group size and computation cost for group key management. By using a hierarchical key distribution architecture and load sharing, the load of key management can be shared by a cluster of third parties without revealing group messages to them. The proposed scheme provides better scalability because the cost for key management of each component is independent of the group size. Specifically, our scheme incurs constant computation and communication overheads for key renewing. In this paper, we present the detailed design of the proposed scheme and performance comparisons with other schemes. Briefly, our scheme provides better scalability than existing group key distribution approaches.

Introduction

In addition to the one-to-one communications, more and more emerging applications involve one-to-many or many-to-many communication patterns. In the group communication model, data packets are transmitted from one or more authorized senders to multiple authorized receivers. Conventional IP multicasting technologies employ group management[1,2,3] and various

Fu-Yuan Lee received the BS degree in computer science from National Chiao Tung University in 1998. He is currently a PhD student in the Department of Computer Science and Information Engineering at National Chiao Tung University. His research interests are in the areas of computer networks and network security.

Shiuhpyng Shieh is a professor and former chairman of Department of Computer Science and Information Engineering of National Chiao Tung University. He is also the director of NCTU-Cisco Internet Technology Center, and the president of Chinese Cryptology and Information Security Association (CCISA), which is the largest and a highly respectable academic organization on information security research in Taiwan. He has worked as advisor to many institutes, such as National Security Bureau, GSN-CERT/CC, National Information and Communication Security Task Force. Before joining NCTU, Dr. Shieh participated in the design and implementation of the B2 Secure XENIX for IBM, Federal Sector Division, Gaithersburg, Maryland. He also designed and developed NetSphinx, a network security product, for Formosoft Inc., which is awarded 1999 network product of the year, Taiwan.

Dr. Shieh received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park. He is a senior member of IEEE, and an editor of Journal of Computer Security, and Journal of Information Science. He was on the organizing committees of numerous conferences, such as ACM conference on Computer and Communications Security, IACR Asiacrypt, Dr. Shieh has published over a hundred academic articles, including papers, patents, and books. Recently he received the Outstanding Research Award from National Chiao Tung University for his academic achievement in research, and the Outstanding Achievement Award from Executive Yuan of Taiwan. His research interests include internetworking, distributed operating systems, and network security.

**Correspondence to: Fu-Yuan Lee, Department of Computer Science and Information Engineering, National Chiao Tung Univserity, Hsinchu, Taiwan 300, China.*
[†]E-mail: leefy@csie.nctu.edu.tw

 Published online 19 March 2004

multicast routing protocols such as, DVMRP,[4] PIM-SM,[5] CBT[6,7,8] and MOSPF[9] to provide multicasting service for multiparty applications. However, securing group communications for dynamic and large-scale multicast groups is relatively unexplored. Clearly, message confidentiality is a very important issue for multicasting.[10,11,12] Cryptographic algorithms are often employed to secure multicast transmissions, where multicast packets are encrypted with cryptosystems. Only legitimate group members with the secret key can acquire the communication contents. In the context of secure group communications, it is important to prohibit the newcomer/ex-member from accessing past/future communications. This requires renewing the secret key that is shared by the group members. For a large group with a highly dynamic membership, the cost of key management, in terms of key establishment and key renewing, can be quite substantial and therefore scalability becomes an important issue that must be addressed.

We can classify the methods for group key management for large dynamic groups into two categories. The first set of schemes typically involves the use of a logical key hierarchy (LKH) which is a set of cryptographic keys organized into a tree structure. On top of the hierarchy is a globally shared common group key and the other keys are employed to assist in the distribution of the common group key.[11,13–18] Consider a multicast group consists of g group members. There are in total $O(g)$ keys and each group member stores $O(\log(g))$ keys. To add or remove a user from the group, a new common group key must be generated and the computation cost for key renewing is $O(\log(g))$.

The second set of schemes decomposes large groups into subgroups. Iolus[19] deals with the scalability issue by partitioning the group members into many subgroups, which are arranged in a hierarchy to create a single multicast group. Scalability is achieved by making each subgroup relatively independent and thus group membership changes can be confined to the respective subgroups. Another essential element that helps Iolus to achieve its scalability is the subgroup agents, which assist in translating messages among subgroups using different subgroup keys. While improving scalability, this approach introduces extra propagation delays and requires full trust in each subgroup agent. In brief, having subgroup agents decrypt and re-encrypt the data packets is a drawback, both from a performance point of view and from a security point of view. IDGKM[20] utilizes the concept of Iolus and a distributed key management architecture to deliver the common group key. It also suffers the drawback of trust in third parties that is necessary for key distribution. SMSDRG[21] proposed a scalable key distribution framework based on the use of cryptographic sequences. Similarly, it requires trust in the third parties and the cost for key distribution is high. DEP[22] uses dual encryption to protect message contents against third parties. However, the cost of communication is substantial because the secret key must be delivered to group members for each packet. SMP[23] provides the same scalability as Iolus but does not need to trust the third parties (the multicast routers). However, the cost of generating secret keys for each data packet is a significant overhead.

This paper proposes a new key distribution scheme for dynamic and large-scale multicast groups. Our scheme distributes the load of the key management to a cluster of third parties without revealing communication contents. As the size of the multicast group grows, new third parties can be included to support the increased computation. In this way, the key management overhead in each component can be kept independent of the group size. This feature enables a single key management server to manage multicast groups with unlimited numbers of group members. Compared with previous approaches, our scheme achieves better scalability in group membership and requires only a constant processing overhead.

Our scheme distributes the load of key management to a cluster of third parties without revealing communication contents.

This paper is organized as follows. In the next two sections, we present the fundamental requirements of a scalable group communication system and describe the functionality of components of a generic secure multicasting framework. We then detail the design of the proposed scheme. Protocol analysis and performance comparison are pre-

sented in the penulfimate section. Finally, we give a brief conclusion in the last section.

Design Considerations

In this section we present several fundamental design considerations for a scalable group key distribution system.

- Only legitimate group members can acquire the communication contents. Third parties involved in key distribution should not have access to the cryptographic keys for encrypting/decrypting communications. A newcomer/ex-member must not be able to derive past/new common group keys.
- For flexibility, key distribution mechanisms should be independent of the underlying multicast routing protocol.
- From the scalability point of view, the cost of each component that assists in key distribution should be independent of the group size.
- No matter how large the group size is, it is required that all the group member can obtain the new common group key in a timely manner. In other words, key distribution must guarantee a certain level of soft real-time property.

Components of Secure Group Communications

In this section, we present a generic framework for secure group communications. The framework is composed of two independent systems: a key distribution system and a multicast data-delivery system. Key distribution systems focus on the construction of a scalable key distribution scheme while multicast data-delivery systems focus on packet transmissions over the multicast backbone. Group members can globally share a common group key via the key distribution scheme and then encrypted packets can be delivered to group members via the multicast data-delivery system.

The multicast data delivery system includes MBone,[24] a virtual network layered upon the Internet for carrying multicast data packets, and various multicast routing protocols running on MBone. MBone is constructed by routers that support routing of multicast data packets. Instead of being physically interconnected, these multicast routers connect to one another via point-to-point tunnels. The subnet directly connected to a multicast route is called a multicast island. Multicast data packets are delivered to multicast islands by multicast routers.

Key distribution systems operate over a *key transporting network* and Figure 1 shows an abstract model of it. As depicted, components involved in the key distribution system are arranged in a hierarchy. The root node is a *key generator*, which is responsible for generating and renewing the common group key. Key generators can be the multicast group creator, one of the group members or a trusted third party. Intermediate nodes, referred as *key distributors*, are network devices or group members with the capability of assisting in the proposed key management operations. Each leaf node represents a subset of group members that attaches to the same key distributor.

In the key-transporting network, each entity is associated with parameters. The key generator maintains parameters of all the other entities and holds secret information, S, for generating the common group key. Each key distributor has a *transformation parameter*, which is assigned by the key generator, and each legitimate group member is associated with a *key deriving key* that is relevant to S and its parent key distributor. In the process of key distribution, the key generator first generates the common group key by using S and a random number as inputs to a key generation function. Instead of sending out the determined common group key, the key generator only delivers the keying materials that enable group members to derive the common group key. Along the paths from the key generator to group members, each key distributor performs a light-weight transformation on the received keying materials and then forwards the result to the next key distributor or respective subgroup members. A legitimate group member who receives the keying materials can subsequently obtain the common group key using its key deriving key and received keying materials.

Group Key Management System

In this section, we first present the procedure for assigning parameters to each component in the

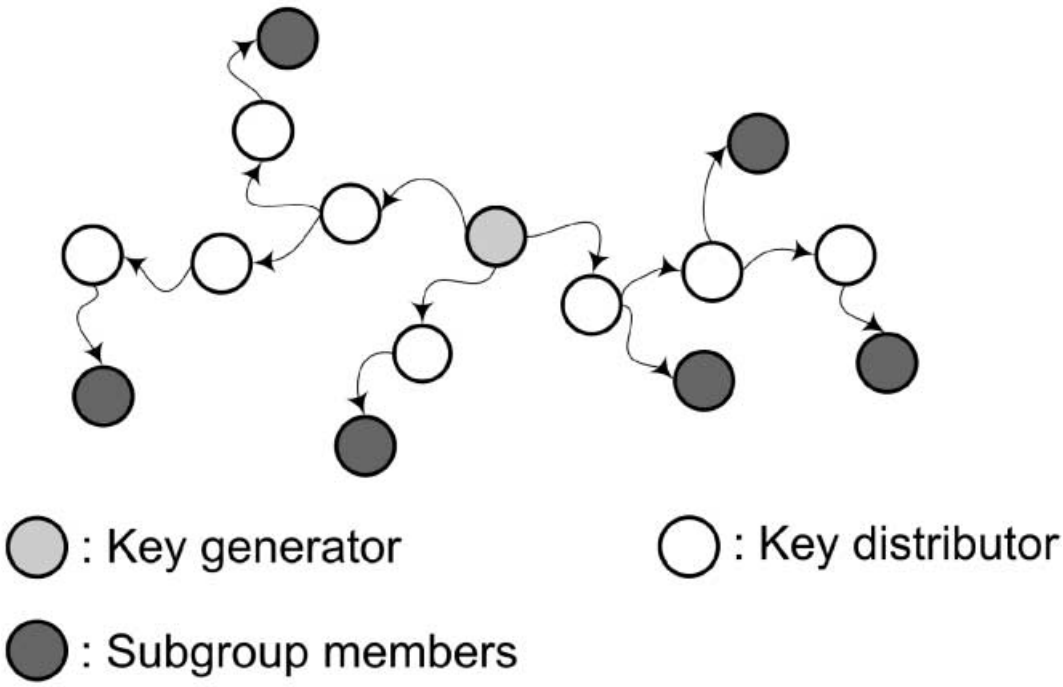

Figure 1. Key transporting network

key-transporting network. Then, we describe the key-renewing procedure. The key-renewing procedure consists of three steps. In the first step, we present the generation of the common group key, explaining how it is forwarded and how group members derive it. The next two steps show the ways of reconfiguring the key-transporting network to ensure forward and backward privacy respectively.

—System Configuration—

Consider a key transporting network, which consists of a key generator and m key distributors. We use N_0 to represent the key generator and N_1, N_2 . . . , N_m for the m key distributors. System configuration of the key transporting network is described as follows.

1. The key generator computes a large integer $n = p*q$, where p and q are carefully chosen large prime numbers. The value of n is publicly available while p and q should be kept secret. Then, the key generator chooses a secret number $S \in Z_n^*$. Moreover, it also selects $m + 1$ distinct numbers, which are denoted as $\{x_0, x_1, x_2, \ldots, x_m\}$ from $Z_{\phi(n)}^*$.
2. Each key distributor holds a transforming parameter that is securely assigned by the key generator. Let $N_{u>0}$ denote a key distributor, N_v denote its parent node and t_u denote the transforming parameter of N_u, where t_u equals $x_u * x_v^{-1} mod(w * \phi(n))$ and w is a large integer with bit length twice as long as n. The key generator sends t_u and $w * \phi(n)$ to N_u securely.
3. Each legitimate group member holds a key-deriving key that is relevant to its point of attachment in the key-transporting network. More precisely, members attached to the same key distributor have the same key-deriving key. Let U_{n_i} represent the set of members attached to key distributor N_i, and K_{n_i} represent the key-deriving key to be assigned to members of U_{n_i}. K_{n_i} equals $S^{x_i^{-1}} mod(n)$.

Figure 2 shows an example of the key-transporting network with configuration parameters.

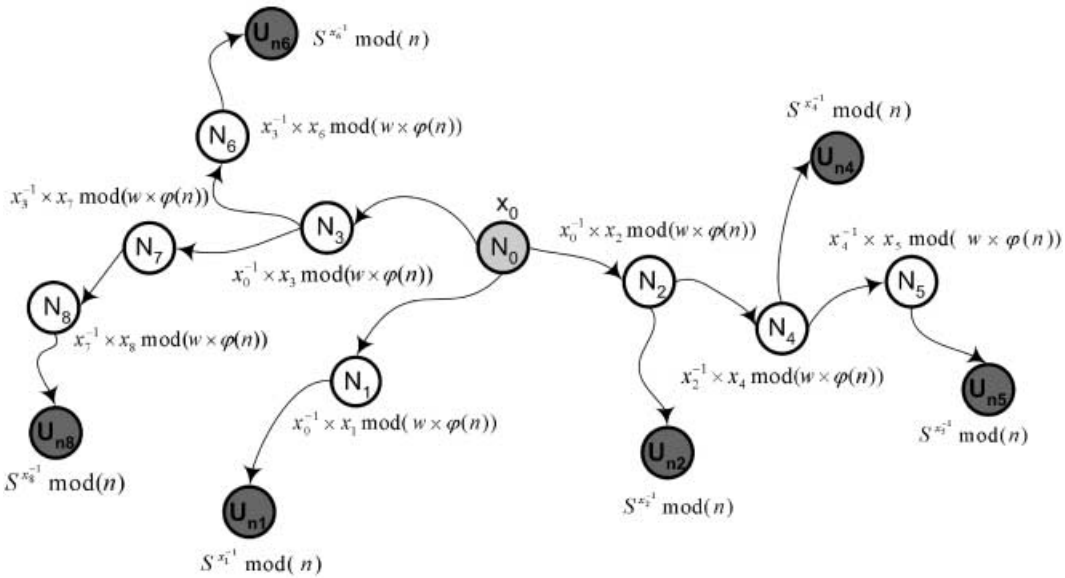

Figure 2. Key transporting network with keys

—Key Distribution—

Key distribution establishes a globally shared common group key for secure group communications. Instead of transmitting the determined common group key in the key transporting network, only the parameters for deriving the common group key are delivered. Along the path from key generator to legitimate group members, each key distributor performs a transformation on the received data and forwards the result to the next key distributor and subgroup members. Ultimately, each legitimate group member derives the common group key by using the key-deriving key and the keying materials received from the parent key distributor as inputs to the key deriving function.

Each legitimate group member derives the common group key by using the key-deriving key and the keying material received from the parent key distributor.

Key generation—The key generator first generates a random number r and computes a pseudo common group key: $PK = S^r \bmod(n)$. After that, the key generator encrypts its identity using PK and signs the cipher text using its private key. Here we assume that the public key of the key generator is publicly available from the public-key infrastructure. We use *authenticator* to represent this signature produced by the generator. Then, the key generator delivers $\{r * x_0 \bmod (w * \phi(n)), authenticator\}$ to the directly connected key distributors.

Forwarding —After receiving the keying materials from the parent key distributor, each key distributor subsequently delivers $\{D * t_i \bmod(w * \phi(n)), authenticator\}$ to its child key distributors and subgroup members, where D represents the received data and t_i is the transformation parameter of the key distributor.

Key derivation—Consider that a group member, A, receives keying materials, denoted as D, from its parent key distributor N_v. We assume that there are k key distributors along the path from the key generator to the group member A. Let $N_{p0} = N_0$, and $N_{p1}, N_{p2}, \ldots, N_{pk}$ denote the k key distributors. Note that $N_{pk} = N_v$. Then, D is equal to:

$$\left\{(r * x_0 mod(w * \phi(n))) * \prod_{i=1}^{k} (x_{p(i-1)})^{-1} * x_{pi} mod(w * \phi(n))\right\} mod(w * \phi(n))$$

Then, A can generate the pseudo common group key, PK, by following the procedure:

$$PK = (K_{nv})^D mod(n) \quad (1)$$

To calculate Equation 1, we observe D and find that:

$$\begin{aligned} D\ mod(\phi(n)) &= \left\{[r * x_0 mod(w * \phi(n))] * \prod_{i=1}^{k}((x_{p(i-1)}^{-1} * x_{pi}) mod(w * \phi(n)))\right\} \\ &\quad mod(\phi(n)) \\ &= \left\{r * x_0 * x_0^{-1} * \left[\prod_{i=1}^{k-1}(x_{pi}^{-1} * x_{pi}) * x_{pk}\right]\right. \\ &\quad mod(\phi(n)) \\ &= r * x_0 * x_0^{-1} * \left[\prod_{i=1}^{k-1}(x_{pi}^{-1} * x_{pi})\right. \\ &\quad \left.mod(\phi(n))\right] * x_{pk} mod(\phi(n)) \\ &= \left(r * 1 * \left[\prod_{i=1}^{k-1} 1\right] * x_{pk}\right) mod(\phi(n)) \\ &= (r * x_{pk}) mod(\phi(n)) \\ &= (r * x_v) mod(\phi(n)) \end{aligned} \quad (2)$$

According to Equation 2 and Euler's generalization of Fermat's little theorem:

$$\begin{aligned} (1) &= \left(S^{x_v^{-1}} mod(n)\right)^{r * x_v} mod(n) \\ &= \left(\left(S^{x_v^{-1}}\right)^{r * x_v}\right) mod(n) \\ &= \left(S^{x_v^{-1} * r * x_v}\right) mod(n) \\ &= S^r\ mod(n) \end{aligned}$$

After obtaining the pseudo common group key, group members first authenticate the derived pseudo common group key with the authenticator. If the verification is successful, group members can subsequently apply a pre-defined function to the pseudo common group key to obtain a secret key with appropriate length that can be used in the chosen symmetric cryptosystem.

—System Reconfiguration for the Join Event—

This section presents the procedure for dealing with the join event. When a host A wants to join the multicast group, it must send the registration request to the key generator first. If the request is granted, the key generator replies with a message containing the key-deriving key and the address of a key distributor. At the same time, the key generator reconfigures the key transporting network prior to executing the key distribution described in the previous section. Let N_v denote the key distributor assigned to the new user and U_{nv} denote the set of subgroup members that the new member will join. The key generator executes the following procedure to reconfigure the key transporting network.

1. Recall that we have chosen $(N + 1)$ integers from $Z^*_{\phi(n)}$. Now, the key generator replaces x_v with x'_v, where $x'_v \in Z^*_{\phi(n)}$.
2. The key generator updates the transformation parameter of N_v:
$$(t_v)_{new} = ((t_v)_{old} * x_v^{-1} * x'_v) mod(w * \phi(n)).$$
3. For key distributors that are children nodes of N_v, the key generator updates their transformation parameters as well. Let N_i denote a child node of N_v, the key generator assigns $(t_i)_{new} = x_i * (x'_v)^{-1}\ mod(w * \phi(n))$ as a new transformation parameter.
4. The key generator securely sends the new key-deriving key $(K_{nv})_{new} = S^{(x'_v)^{-1}}\ mod(n)$ to the new group member.
5. To enable the members of $(U_{nv})_{old}$ to derive the new key-deriving key, the key generator sends $S^{(x'_v)^{-1} - x_v^{-1}}\ mod(n)$ to N_v first. Then, N_v forwards it to all local members securely. This can be achieved by utilizing one-to-one secure communications or other secure-group communications for a small multicast group. Now, group members that attach to N_v can derive the new key-deriving key:
$$\begin{aligned} (K_{nv})_{new} &= \left[K_{nv} * \left(S^{(x'_v)^{-1} - x_v^{-1}} (mod(n))\right)\right] mod(n \\ &= S^{x_v^{-1}} * S^{(x'_v)^{-1} - x_v^{-1}} mod(n) \\ &= S^{(x'_v)^{-1}} mod(n) \end{aligned}$$

When the reconfiguration completes, the key generator can now execute the key distribution to renew the common group key.

—System Reconfiguration for the Leave Event—

Reconfiguring the key-transporting network for a leave event is similar to the procedure for a join event. There are two differences:

- Step 4 is deleted.
- In step 5, N_v must forward the information received from the key generator to the members of U_{nv} via one-to-one secure communications.

—Management of Key Transporting Network—

So far we have used the key transporting network without considering how it is constructed and managed. However, in practice, the scalability offered in the proposed scheme highly depends on the structure of the key transporting network. As mentioned above, key distributors can be group members or third parties and the only requirement is the capability to perform transformation and forwarding procedures. From this point of view, multicast routers would be appropriate candidates for key distributors. One naive configuration is that every multicast router is a key distributor and each multicast router manages the group members in their respective multicast islands only. This configuration is appropriate if every multicast island has sufficient group members. If group members are sparsely distributed, it is better to use one key distributor to manage members of several nearby multicast islands. It is worthy of note that it might also happen that all of the group members attached to a key distributor leave the multicast group after a period of time. This would result in a huge but sparse key transporting network. To avoid this, the key generator should periodically delete or combine key distributors to increase compactness and efficiency.

To avoid a huge but sparse key-transporting network, the key generator should periodically delete or combine key distributors to increase compactness and efficiency.

Protocol Analysis and Performance Comparison

We have presented a scalable group key management system that supports large and dynamic multicast groups. This section shows how the proposed system meets the security requirements of secure-group communications and compares the proposed system with other schemes.

—Protocol Analysis—

According to the requirements described above, our scheme can be examined in following aspects:

Data confidentiality—Outsiders and intermediate nodes such as key distributors or multicast routers are unable to access the communication contents. To acquire the contents, attackers must either hold the secret key or perform brute force attack on the encrypted multicast packets. Brute force attacks can be easily impeded by using cryptographic algorithms with sufficient key length. Obtaining the secret key requires key-deriving keys, which are held by legitimate group members only. Therefore, only legitimate group members can acquire communication contents.

Backward and forward secrecy—A new key-deriving key cannot be used for generating either the old key-deriving key or the old common group key. As a result, past transmissions are not accessible to new members. Likewise, an ex-member with an old key-deriving key cannot derive the common group key that is used to encrypt and decrypt current transmissions.

Processing scalability—The amount of processing of each component to cope with key management and secure transmissions is independent of the group size. In terms of secure transmission, the size of the encrypted message only depends on the size of the original message and the cost of encryption/decryption is irrelevant to the number of group members. In key distribution, the number of messages transmitted by a node (key generator or key distributor) does not depend on the group size but depends only on the number of child nodes attached.

Membership scalability—The proposed key-management system employs a distributed approach to achieving key distribution and the cost of each component for key distribution is independent of the group size. Specifically, the

cost of key renewing is independent of the group size and therefore our scheme is scalable in terms of membership.

Loss of key-update handling—If a group member does not receive the key-update message, the group member sends the request for retransmission to its key distributor instead of to the key generator. In this way, when numerous loss of key update events occur in the same time, retransmissions of keying materials can be processed concurrently and in a distributed manner.

—Performance Comparison—

We evaluate the performance of our proposed scheme on both communication and computation complexities. The major measure of communication complexity is the bandwidth consumption, e.g., the number of messages as well as the size of the messages, for transmitting key update messages. Computation complexity is concerned with the storage requirement of all group members, the key generator and the key distributors, and the operations for deriving/generating/forwarding the key.

In the proposed scheme, the number of bits in every key-update message is less than $log(w * \phi(n))$, which is independent of the size of the group. As to the number of messages, the key generator sends two messages for reconfiguration and d messages to the directly connected key distributors for key distribution, where d represents the number of child nodes connected to the key generator. As a result, the total number of message is $(2 + d)$, which is independent of the size of the group. Similarly, each key distributor sends d messages to downstream key distributors. In addition, the number of messages for securely distributing keying materials to subgroup members is only relevant to the number of subgroup members attached to the key distributor. From the storage point of view, the key generator must manage the parameters in the key-transporting network. Thus, it is proportional to the number of nodes in the key-transporting network. Each key distributor is required to store the transformation parameters only and each group member must store two secret keys, one is the common group key and the other is the key-deriving key. With regard to the operations, one exponential modular operation is used in key generation and key derivation respectively and one multiply modular operation is used in key forwarding.

Table 1 compares our scheme with other scalable, secure multicasting protocols. Here we define the symbols used in the table. We use g to denote the number of members in the multicast group. For schemes using the subgrouping technique, m

		IOLUS	DEP	LKH	SMP	SMSDRG	SLKD
Total no. of keys		O(g)	O(g)	O(g)	O(g)	O(g)	O(g)
No. of keys per member		2	4	O(log(g))	2	2	2
No. of keys at an subgroup agent		2	2	—	2	1	1
No. of keys at key management server		2	$c + 2$	O(g)	O(m)	O(m)	O(m)
Cost of a JOIN event	non-requesting member	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)
	requesting member	O(1)	O(1)	O(log(g))	O(1)	O(1)	O(1)
	key management server	O(1)	O(c)	O(log(g))	O(1)	O(s)	O(1)
	involved subgroup agent	O(1)	O(1)	—	O(1)	O(1)	O(1)
	non-involved subgroup agent	O(1)	O(1)	—	O(1)	O(1)	O(1)
	no. of key encryptions	O(m)	O(m)	O(log(g))	1	O(m)	0
Cost of a LEAVE event	non-requesting member	O(1)	O(1)	O(log(g))	O(1)	O(1)	O(1)
	key management server	O(1)	O(c)	O(log(g))	O(1)	O(s)	O(1)
	involved subgroup agent	O(s)	O(s)	—	O(s)	O(1)	O(s)
	non-involved subgroup agent	O(1)	O(1)	—	O(1)	O(1)	O(1)
	no. of key encryptions	O(m)	O(m)	O(log(g))	1	O(m)	0
Need to trust intermediate nodes		Yes	No	—	No	Yes	No

Table 1. Comparison

represents the number of subgroups, *s* denotes the average subgroup size and *c* represents the number of key groups. Moreover, we use SLKD to denote the scheme presented in this paper. For subgroup-based schemes, 'subgroup agent' represents the role that manages the subgroup in each scheme respectively.

Table 1 shows that our scheme, SLKD, provide better scalability then existing schemes. It is clear that SLKD incurs less computational overheads than LKH-based schemes in terms of all performance metrices. In contrast to existing schemes, SLKD does not require group members to trust third parties and does not need an encryption operation in key distribution.

Conclusion

In this paper, we proposed a scalable, group-key distribution scheme designed to support secure communications in large dynamic multicast groups. In the proposed key-transporting network, key distributors share the load of key management with the key generator. When the multicast group grows, new key distributors can be included to support the management overhead incurred by new group members. As a result, in each component the cost of performing key management is independent of the group size. Although key distributors are used to distribute keying materials, our approach protects message privacy against them. Only legitimate group members can have access to the secure message contents. Therefore, our scheme has better scalability than existing schemes.

Acknowledgement

This work is supported in part by Lee and MTI Center for Interworking Research (Global Crossing). Ministry of Education and National Science Council under contract NSC 92-2213-E-009-093-, NSC 92-2213-E-009-122-, NSC 92-2219-E-009-002-.

References

1. Deering SE. Multicast Routing in Internetworks and Extended LANs, in *Proceedings ACM SIGCOMM*, Aug. 1988; 55–64.
2. Deering S. Host Extensions for IP Multicasting, Aug. 1989; *RFC-1112*.
3. Fenner W. Internet Group Management Protocol, Version 2, Nov. 1997; *RFC-2236*.
4. Waitzman D, Partridge C, Deering S. Distance Vector Multicast Routing Protocol, Nov. 1988; *RFC-1075*.
5. Estrin D, Farinacci D, Helmy A, Thaler D, Deering S, Handley M, Jacobson V, Liu C, Sharma P, Wei L. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, Jun. 1998; *RFC-2362*.
6. Ballardie A. Core Based Trees (CBT) Multicast Routing Architecture, Sep. 1997; *RFC-2201*.
7. Ballardie A. Core Based Trees (CBT version 2) Multicast Routing–Protocol Specification, Sep. 1997; *RFC-2189*.
8. Ballardie T, Francis P, Crowcroft J. Core based trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing, *Proceedings ACM SIGCOMM*, 1993; **23**: No. 4, 85–95.
9. Moy J. Multicast Extensions to OSPF, Mar. 1994; *RFC-1584*.
10. Bagnall P, Briscoe R, Poppitt A. Taxonomy of Communication Requirements for Large-scale Multicasting Applications, Dec. 1999; *RFC-2729*.
11. Wallner D, Harder E, Agee R. Key Management for Multicast: Issues and Architectures, Jun. 1999; *RFC-2627*.
12. Moyer MJ, Rao JR, Rohatgi P. A Survey of Security Issues in Multicast Communications, *IEEE Network*, Nov./Dec. 1999; **13**: 12–23.
13. Wong CK, Gouda M, Lam SS. Secure Group Communications Using Key Graphs, *IEEE/ACM Transaction Networking*, Feb. 2000; **8**: 16–30.
14. Waldvogel M, Caronni G, Sun D, Weiler N, Plattner B. The VersaKey framework: versatile group key management, *IEEE Journal Select. Areas Communication*, Sep. 1999; **17**: 1614–1631.
15. Perrig A, Song D, Tygar J. ELK, a New Protocol for Efficient Large-Group Key Distribution, in *IEEE Symposin Security and Privacy*, May 2001; 247–262.
16. Wong CK, Lam SS. Keystone: A Group Key Management Service, in *International Conference. Telecommunications*, 2000.
17. Setia S, Koussih S, Jajodia S, Harder E. Kronos: A Scalable Group Re-Keying Approach for Secure Multicast, in *IEEE Symposin Security and Privacy*, 2000; 215–228.
18. Li XS, Yang YR, Gouda MG, Lam SS. Batch rekeying for secure group communications, in *Proceedings 10th international Conference World Wide Web*, 2001; 525–534.
19. Mittra S. Iolus: a framework for scalable secure multicasting, in *Proceedings ACM SIGCOMM.*, 1997; 277–288.

20. Hardjono T, Cain B. Secure and Scalable Inter-Domain Group Key Management for N-to-N Multicast, in *Proceedings International Conference Parallel and Distributed Systems*, December 1998.
21. Molva R, Pannetrat A. Scalable multicast security with dynamic recipient groups, *ACM Transaction Information and System Security*, 2000; **3**: No. 3, 136–160.
22. Dondeti L, Mukherjee S, Samal A. A Dual Encryption Protocol for Scalable Secure Multicasting, in *IEEE Symposin Computer and Communications*, 1999; 2–8.
23. Yang WH, Fan KW, Shieh SP. A secure multicast protocol for the Internet's multicast backbone, *ACM/PH International Journal Network Management*, 2001; **11**: 129–136.
24. Eriksson H. MBONE: The Multicast Backbone, *ACM Communication*, August 1994; **37**: 54–60. ■

