A Scalable Multicast Security Protocol in Hierarchy Structures

CHIN-CHEN CHANG[1,2], JUNG-SAN LEE[2], YA-FEN CHANG[3] and YI-PEI HSIEH[2]
[1] Department of Information Engineering and Computer Science, Feng Chia University,
Taichung, Taiwan, 40724, R.O.C.
ccc@cs.ccu.edu.tw
[2] Department of Computer Science and Information Engineering, National Chung Cheng University,
Chiayi, Taiwan, 621, R.O.C.
{ljs, hsiehyp}@cs.ccu.edu.tw
[3]Department of Logistics Engineering and Management,
National Taichung Institute of Technology, Taichung 404, Taiwan, R.O.C.
cyf@cs.ccu.edu.tw

Abstract: - The fundamental function of the network security protocols is to allow the authorized participant to communicate with others securely over the insecure network. Point-to-point packet transmission is the most common way of the communication over the network. However, it is not appropriate for many emerging applications such as message services, pay-TV, teleconference, or collaborate tasks. It is due to that these applications are structured on the group communication. Consequently, point-to-group or group-to-group packet transmission has become an important issue of the network in recent periods. In 2004, Aslan proposed a scalable multicast security protocol using a subgroup-key hierarchy. Aslan's protocol allows the user to communicate with others efficiently. Nevertheless, we find that each communicating user in the system has to maintain many secret keys such that it is not convenient for all users. Besides, while a member joins or leaves the communicating group, lots of involved participants have to change their secret keys to confirm the forward secrecy and the backward secrecy. In this article, we propose an improved multicast security protocol which not only preserves the functionality of Aslan's protocol but also possesses the performance better than other related works.

Key-Words: - multicast communication, scalability, secret key, hierarchy

1 Introduction

Engineers have proposed lots of security protocols for providing secure communications for network users recently. Among them, those of the multicast communications are regarded as the most critical ones. It is due to that more and more emerging applications, such as teleconference, pay-TV, collaborating tasks and message services, are based on the group communication. Traditional point-to-point communications do not suffice for users' needs anymore. Point-to-group and group-to-group communications have become the important research issue in computer networks [2, 3, 4, 5, 7, 8, 10, 11].

So far, there are three main solutions, central control, distributed control and subgroup control, proposed for providing secure multicast communications and key distribution.

Central control: A central manager takes the responsibility for the security of the whole group and key distribution. However, this solution is not scalable for large groups. Besides, the failure of the central manager will make the communication of the whole group inactive [6].

Distributed control: All group members take the responsibility for key generation and the security of the whole group. Although this solution can get rid of the disadvantage of the central control solution, in which the failure of the central manager will make the group communication inactive, the scalability of the distributed control solution for large groups is worse than that of the central control one [4, 12, 13].

Subgroup control: The whole group is divided into several subgroups. Each subgroup is controlled by a subgroup manager. The scalability of this approach is better than those of the above two solutions. Furthermore, the failure of the single subgroup manger does not lead to the inactivity of the whole group communication [1, 9].

In 2004, Aslan presented a scalable multicast security protocol with the subgroup control solution using a hierarchy structure. Aslan's protocol makes all group members be able to communicate with others efficiently. However, we find that there exist two weaknesses in her multicast protocol. First, each group user has to keep lots of secret key. This is inconvenient for the involved participants. Second, while a member joins or leaves the group, lots of involved participants have to change their secret keys to confirm the forward secrecy and the backward secrecy. We consequently propose an improved version which not only preserves the functionality of Aslan's protocol but also has the performance better than that of other schemes.

The rest of this paper is organized as follows. A review of Aslan's protocol is described in Section 2, followed by our proposed multicast security protocol in Section 3. Discussions of our protocol and the comparisons between other related works and our scheme are shown in Section 4. Finally, we make some conclusions in Section 5.

2 Related works

Based on a hierarchy structure, Aslan proposed a scalable multicast security protocol with the subgroup control solution [1]. The whole structure of Aslan's protocol is illustrated in Fig.1.

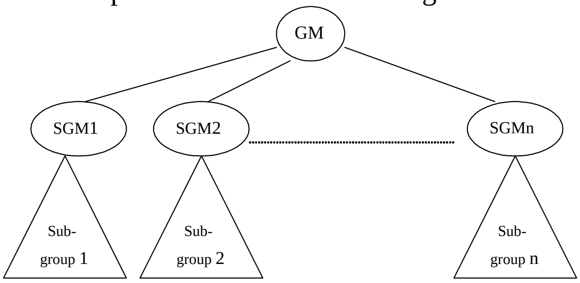

Fig.1: the structure of Aslan's protocol

The main idea of his protocol is to divide the whole group into several subgroups. As shown in Fig.2, each subgroup *i* is formed with a hierarchy structure and is controlled by a subgroup manager SGM*i*, where *i* = 1, 2, …, n and n is the number of subgroups. In Aslan's protocol, each subgroup user has to keep many secret keys. For example, in Subgroup *i*, the user U_1 has to store K*i*(h, 1), K*i*(h-1, 1), …, K*i*(0, 1), where h is the height of Subgroup *i*, and d is the maximum degree of each internal nodes. That is, each user must keep all secret keys of the path from the subgroup manager to himself/herself. The group manager GM shares a different secret key K(GS*i*) with each SGM*i*. Besides, GM generates another secret key K(GS) shared between all SGM*i*'s.

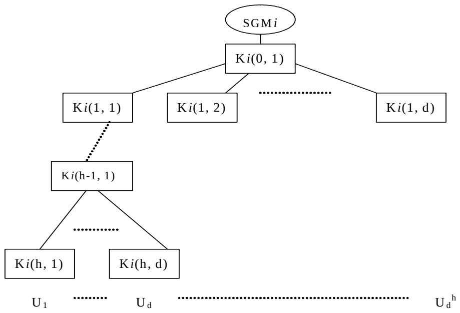

Fig.2: the structure of Subgroup *i* in Aslan's protocol

Furthermore, several assumptions are made in the multicast system. First, the number of subgroups increases progressively. That is, SGM*i*'s are not permitted to leave the group. Second, while a new member wants to join the group, GM must take the responsibility for finding an empty place for him/her. If all subgroups are full, GM has to create a new subgroup. Third, all SGM*i*'s and group members are assumed to be trustworthy.

There are four main operations in Aslan's protocol: subgroup manager join operation, message broadcast operation, member join operation and member leave operation. The details of these operations are described as follows.

Subgroup manager join operation:

If a new subgroup manager SGM_{n+1} joins the group, GM has to change the secret key K(GS) to $K_{new}(GS)$. Besides, GM generates a new secret key $K(GS_{n+1})$ shared between GM and SGM_{n+1}. Next, GM computes

$$E_{K(GS)}[K_{new}(GS)] \text{ and}$$

$$E_{K(GS_{n+1})}[K_{new}(GS)],$$

where $E_k[m]$ is the encryption algorithm to encrypt the message m with the secret key k. GM then broadcasts the computation results to all SGM*i*'s including the new one. While receiving the messages, the original SGM*i*'s retrieve the new secret key $K_{new}(GS)$ by computing

$$D_{K(GS)}[E_{K(GS)}[K_{new}(GS)]],$$

where $D_k[m]$ is the decryption algorithm to decrypt the message m with the secret key k. On the other hand, the new subgroup manager retrieves the new secret key $K_{new}(GS)$ by computing

$$D_{K(GS_{n+1})}[E_{K(GS_{n+1})}[K_{new}(GS)]].$$

Hence, the join operation of a new subgroup manager is completed.

Message broadcast operation:

While a message M needs to be broadcasted, GM generates a new secret key K(msg) to encrypt M. GM then computes the followings,

$E_{K(msg)}[M]$ and

$E_{K(GS)}[K(msg)]$,

and broadcasts the computation results to all SGM*i*'s. After receiving the messages, each SGM*i* computes

$K(msg) = D_{K(GS)}[E_{K(GS)}[K(msg)]]$ and

$E_{Ki(0,\,1)}[K(msg)]$,

where K*i*(0, 1) is the common secret key shared between all members in Subgroup *i*.

SGM*i* then broadcasts the two following messages to all subgroup users:

$E_{Ki(0,\,1)}[K(msg)]$ and

$E_{K(msg)}[M]$.

Therefore, each subgroup user can obtain the message as follows:

$K(msg) = D_{Ki(0,\,1)}[E_{Ki(0,\,1)}[K(msg)]]$ and

$M = D_{K(msg)}[E_{K(msg)}[M]]$.

Member join operation:

While a new user U_d wants to join the communication group, GM has to find an appropriate place and generate a secret key $Ki_{new}(h,\ d)$ for U_d. As illustrated in Fig.3, all secret keys of the path from SGM*i* to U_d must be modified to preserve the backward secrecy. The secret key K*i*(*j*, 1) will be changed to $Ki_{new}(j,\ 1)$, where *j* = 0, 1, …, h-1. These new secret keys have to be sent to the involved participants securely. Hence, SGM*i* computes and broadcasts the following messages to all subgroup members.

$E_{Ki(0,\,1)}[Ki_{new}(0,\ 1)]$,

$E_{Ki(1,\,1)}[Ki_{new}(1,\ 1)]$,

M

$E_{Ki(h-1,\,1)}[Ki_{new}(h-1,\ 1)]$ and,

$E_{Ki(h,\,d)}[Ki_{new}(0,\ 1),\ Ki_{new}(1,\ 1),\ \ldots,\ Ki_{new}(h-1,\ 1)]$.

$Ki_{new}(0,\ 1)$ is required by all subgroup members, $Ki_{new}(1,\ 1)$ is required by the first d^{h-1} members, and so forth until $Ki_{new}(h-1,\ 1)$ is only required by the first d subgroup members. Furthermore, U_d will obtain all needed secret keys by computing

$$D_{Ki(h,\,d)}[E_{Ki(h,\,d)}[Ki_{new}(0,\ 1),\ Ki_{new}(1,\ 1),\ \ldots,\ Ki_{new}(h-1,\ 1)]].$$

Here, member join operation is completed.

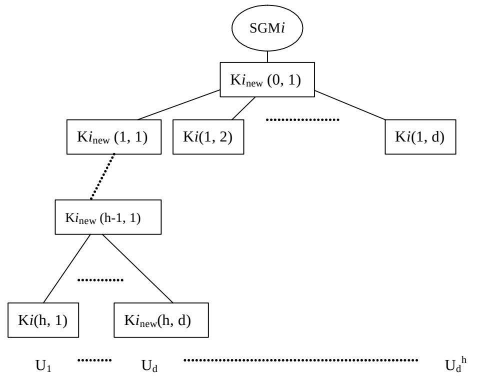

Fig.3: the example of member join operation in Aslan's protocol

Member leave operation:

While a user U_1 leaves Subgroup *i*, as illustrated in Fig.4, all secret keys of the path from SGM*i* to U_1 must be modified to preserve the forward secrecy. SGM*i* has to compute and broadcast the following messages to all subgroup members.

$E_{Ki(1,\,f)}[Ki_{new}(0,\ 1)]$,

$E_{Ki(2,\,f)}[Ki_{new}(1,\ 2)]$,

M

$E_{Ki(h,\,f)}[Ki_{new}(h-1,\ 1)]$ and,

$E_{Ki_{new}(h-1,\,1)}[Ki_{new}(0,\ 1),\ Ki_{new}(1,\ 1),\ \ldots,\ Ki_{new}(h-2,\ 1)]$,

where *f* = 2, 3, …, d. Hence, all involved participants can obtain the required secret keys securely. That is, member leave operation is completed.

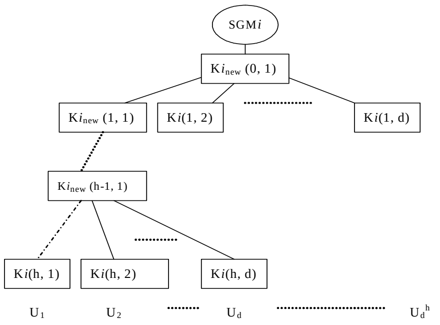

Fig.4: the example of member leave operation in Aslan's protocol

3 The Improved Multicast Security Protocol

Although Aslan's multicast protocol is efficient and scalable, members in the multicast system have to keep lots of secret keys. Furthermore, the operations of member join and member leave make many involved participants have to change their required secret keys to confirm the forward secrecy and the backward secrecy. Hence, we propose an improved version with Lagrange Interpolating Polynomial. Instead of maintaining lots of secret keys, each member in the multicast system needs to keep only one secret key. Besides, a bulletin board is adopted in our protocol. Note that only GM and the legal SGM*i*'s can modify and update the bulletin board.

The same as Aslan's scheme, the whole group is divided into several subgroups. As shown in Fig.5, each subgroup formed with a hierarchy structure of height h is controlled by a subgroup manager SGM*i*, where *i* = 1, 2, …, n and n is the number of the subgroups. Every node in the hierarchy structure of Subgroup *i* is assigned a unique identity ID*i*(*b*, *j*), where *b* = 0, 1, …, h, *j* = 1, 2, …, d, and d is the maximum degree of the internal node in the hierarchy tree.

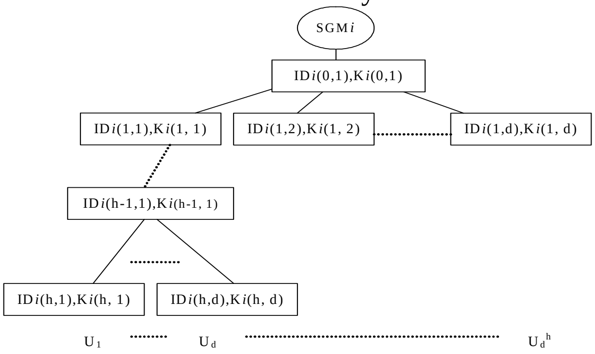

Fig.5: the hierarchy structure of Subgroup *i*

There are four operations in our protocol: subgroup manager join operation, member join operation, member leave operation, and message broadcast operation. The first operation is inherited from Aslan's protocol for its practicability. The notations used in our protocol are the same as those in Aslan's protocol. The details of other three operations are described as follows.

Definition of Lagrange Interpolating Polynomial:

Let (x_1, y_1), (x_2, y_2), …, (x_t, y_t) be *t* points on two-dimensional plane [14]. *N* is a prime. a_0, a_1, …, and a_{t-1} are integers ranged within [1, *N*-1]. We can obtain the polynomial by computing

$$f(x) = \sum_{j=1}^{t} y_j \prod_{i=1, i\neq j}^{t} \left(\frac{x - x_i}{x_j - x_i} \right) \bmod N$$

$$= a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \bmod N,$$

where $y = f(x)$.

The bulletin board setup:

At first, GM takes the responsibility for constructing the bulletin board as shown in Table 1. For each internal node ID*i*(*b*, *j*) in Subgroup *i*, SGM*i* bottom-up computes a corresponding polynomial ID*i*(*b*, *j*)_P(x) as follows, where *b* = h-1, h-2, …, 0 and *j* = 1, 2, …, d.

Step 1: Computes d hash values

h_*ib*1 = h(K*i*(*b*+1, 1), ID*i*(*b*, *j*), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d)),

h_ *ib*2 = h(K*i*(*b*+1, 2), ID*i*(*b*+1, 1),ID*i*(*b*, *j*), ID*i*(*b*+1, 3), ID*i*(*b*+1, 4), …, ID*i*(*b*+1, d)),

⋮

h_ *ib*d = h(K*i*(*b*+1, d), ID*i*(*b*+1, 1), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d-1), ID*i*(*b*, *j*)).

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates (h_*ib*1, K*i*(*b*, *j*)), (h_ *ib*2, K*i*(*b*, *j*)), …, and (h_ *ib*d, K*i*(*b*, *j*)), to obtain the polynomial

ID*i*(*b*, *j*)_P(x) = $a_0 + a_1x + \ldots + a_{d-1}x^{d-1}$,

where a_0, a_1, …, a_{d-1} are integers.

Step 3: Publishes all identities of nodes and their corresponding polynomials on the bulletin board.

Table 1: the example of the bulletin board

SGM*i*	
Node	Polynomial
ID*i*(*b*, 1)	ID*i*(*b*, 1)_P(x)
ID*i*(*b*, 2)	ID*i*(*b*, 2)_P(x)
⋮	⋮
ID*i*(*b*, d)	ID*i*(*b*, d)_P(x)

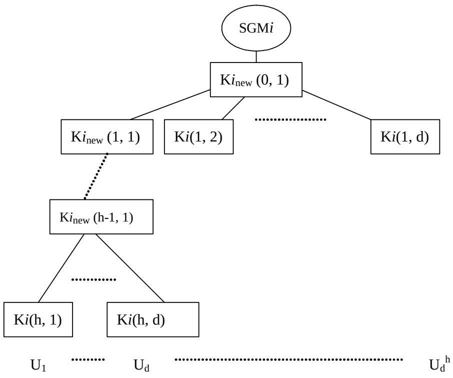

Fig.6: the example of member join operation of our protocol

Member join operation:

While a new member U_d wants to join the communication group, GM has to find a suitable place and generate a secret key for him/her. As illustrated in Fig.6, all secret keys of the path from SGM*i* to U_d must be modified to confirm the backward secrecy. The secret key K*i*(*b*, 1) must be changed, where *b* = 0, 1, …, h-1. All involved internal nodes' polynomials published on the bulletin board will be updated by SGM*i*. That is, SGM*i* has to bottom-up perform

Lagrange Interpolating Polynomial (h-1) times to reconstruct (h-1) involved polynomials. For each involved internal node ID*i*(*b*, 1), where *b* = h-1, h-2, …, 0 (i.e. the internal nodes on the path from U$_d$ to SGM*i*), SGM*i* executes the followings.

Step 1: Computes d hash values

h_*ib*1 = h(K*i*(*b*+1, 1), ID*i*(*b*, 1), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d)),

h_*ib*2 = h(K*i*(*b*+1, 2), ID*i*(*b*+1,1),ID*i*(*b*,1), ID*i*(*b*+1,3), ID*i*(*b*+1,4), …, ID*i*(*b*+1, d)),

M

h_*ib*(d-1)=h(K*i*(*b*+1,d-1),ID*i*(*b*+1,1),ID*i*(*b*+1,2), …, ID*i*(*b*+1,d-1), ID*i*(*b*,1),D*i*(*b*+1,d),

h_ *ib*d = h(K*i*(*b*+1, d), ID*i*(*b*+1, 1), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d-1), ID*i*(*b*, 1)).

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates (h_*ib*1, K*i*(*b*, 1)), (h_ *ib*2, K*i*(*b*, 1)), …, and (h_ *ib*d, K*i*(*b*, 1)), to obtain the polynomial

ID*i*(*b*, 1)_P(x) = $a'_0 + a'_1x + \dots + a'_{d-1}x^{d-1}$,

where $a'_0, a'_1, \dots, a'_{d-1}$ are integers.

Step 3: Updates the modified information on the bulletin board as shown in Table 1.

Member leave operation:

While a user U$_d$ leaves Subgroup *i*, as illustrated in Fig.7, all secret keys of the path from SGM*i* to U$_d$ must be modified to confirm the forward secrecy. The secret key K*i*(*b*, 1) must be changed, where *b* = 0, 1, …, h-1. All involved internal nodes' polynomials published on the bulletin board will be modified by SGM*i*. That is, SGM*i* has to bottom-up perform Lagrange Interpolating Polynomial (h-1) times to reconstruct (h-1) involved polynomials.

For each involved internal node ID*i*(*b*, 1), where *b* = h-1, h-2, …, 0 (i.e. the internal nodes on the path from U$_d$ to SGM*i*), SGM*i* executes the followings.

Step 1: Computes

h_*ib*1 = h(K*i*(*b*+1, 1), ID*i*(*b*, 1), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d-1)),

h_*ib*2 = h(K*i*(*b*+1, 2), ID*i*(*b*+1,1),ID*i*(*b*,1), ID*i*(*b*+1,3), ID*i*(*b*+1,4), …, ID*i*(*b*+1, d-1)),

M

h_ *ib*(d-1) = h(K*i*(*b*+1, d-1), ID*i*(*b*+1, 1), ID*i*(*b*+1, 2), …, ID*i*(*b*+1, d-2), ID*i*(*b*, 1)).

Step 2: Performs Lagrange Interpolating Polynomial on these coordinates (h_*ib*1, K*i*$_{new}$(*b*, 1)), (h_ *ib*2, K*i*$_{new}$(*b*, 1)), …, and (h_ *ib*(d-1), K*i*$_{new}$(*b*, 1)), to obtain the polynomial

ID*i*(*b*, 1)_P(x) = $a''_0 + a''_1x + \dots + a''_{d-1}x^{d-2}$,

where $a''_0, a''_1, \dots, a''_{d-1}$ are integers.

Step 3: Updates the modified information on the bulletin board.

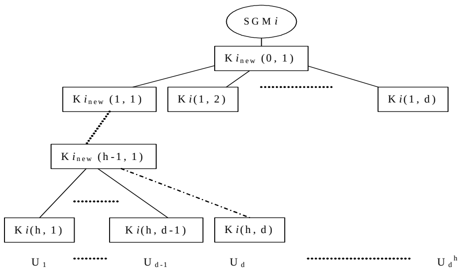

Fig.7: the example of member leave operation of our protocol

Message broadcast operation:

While a message M needs to be broadcasted, GM generates a new secret key K(msg) to encrypt M. Next, GM computes the followings,

$E_{K(msg)}$[M] and

$E_{K(GS)}$[K(msg)],

and then broadcasts the computation results to all SGM*i*'s. After receiving the messages, each SGM*i* computes

K(msg) = $D_{K(GS)}[E_{K(GS)}$[K(msg)]] and

$E_{Ki(0, 1)}$[K(msg)],

where K*i*(0, 1) is the common secret key shared by all members in Subgroup *i*.

SGM*i* then broadcasts the following messages to all subgroup users,

$E_{Ki(0, 1)}$[K(msg)] and

$E_{K(msg)}$[M].

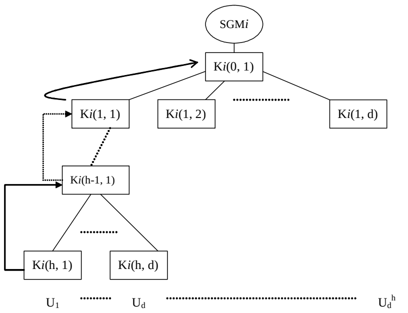

Fig.8: the example of message broadcast operation in Subgroup *i*

As shown in Fig.8, the subgroup member U$_1$ uses his/her secret key to obtain the secret key of the upper level by computing

K*i*(h-1, 1) = ID*i*(h-1, 1)_P(h_*ib*1),

where h_*ib*1 = h(K*i*(h, 1), ID*i*(h-1, 1), ID*i*(h, 2), ID*i*(h, 3), …, ID*i*(h, d)) is pre-computed by U$_1$ and *b* = h-1. By the same way, U$_1$ can quickly obtain K*i*(0, 1) to retrieve K(msg) and decrypt the message M.

4 Discussions

In the following, several comparisons between other related works and our proposed scheme are presented. At first, we define the notations used in Table 2.

n: the total number of subgroups
m: the total number of subgroup members
d: the maximum degree of each internal node in the hierarchy structure
h: the height of each subgroup, i.e. $m = d^h$
$h_{1:}$ $h_1 = \log_d (n * m)$
SGM: the subgroup manager
in: the involved member
non: the non-involved member
K: the symmetric en/decryption operation
P: the operation of constructing the polynomial by Lagrange Interpolating Polynomial
Y: the operation for obtaining y with input x, where y = f(x)

Obliviously, the computation load of constructing the polynomial by Lagrange Interpolating Polynomial is quite lighter than that of performing the symmetric en/decryption operation. It is due to that the construction of LIP polynomial is based on simple multiplication while that of en/decryption function such as DES and FEAL, depends on lots of round operations including permutation, key transformation, expansion, substitution, and modulus [15-19]. As shown in Table 2, in the case for a new member to join the communication group, the computation loads of each subgroup manager in our protocol are lighter than those in Aslan’s and Wong et al.’s protocol but similar to those in Iolus protocol. As for member join operation, the computation loads of each subgroup member in our protocol are quite lighter than those in other related works.

Table 2: the comparisons between other related works and our protocol

Schemes Operations		Ours	Aslan’s	Iolus[9]	Wong et al.’s[6]
Member join	SGM	(h-1)*P+ (h*d)*H	2h*K	2K	$2h_1$*K
	in	0	h*K	1K	h_1*K
	non	0	(d/d-1)*K	1K	(d/d-1)*K
Member leave	SGM	(h-1)*P+ (h*d)*H	[2(hd-1)+(h-d-1)]*K	(m-1)*K	[2(h_1d-1)+(h_1-d-1)]*K
	in	-	-	-	-
	non	0	(3d/d-1)*K	1K	(3d/d-1)*K
Message broadcast	SGM	-	-	-	-
	in	h*Y+1K	1K	1K	1K
	non	-	-	-	-

Furthermore, in case that a member leaves the communication group, the computation loads of both subgroup managers and subgroup members in our protocol are far lighter than those in other related works. Nevertheless, involved users of our scheme have to perform h*Y+1K operations for retrieving a broadcast message while those of other works only need to execute 1K operation. We therefore conclude that our scheme is more suitable for the multicast system with high mobility than other related works.

5 Conclusion

Aslan proposed a scalable multicast security protocol for providing secure communications with a hierarchy structure in 2004. However, we find that each subgroup member in the multicast system has to keep many secret keys in his/her databases. This is so inconvenient for all users. Furthermore, while a member joins or leaves the communicating group, lots of involved participants have to modify their secret keys to confirm the forward and backward secrecies. We thus present an improved version based on the structure of Aslan’s scheme. Each group member in our multicast system needs to keep only one secret key in his/her database. As shown in Table 2, our proposed protocol not only preserves the functionality of Aslan’s protocol but also possesses the performance better than that of other related works. Specifically, our scheme is more suitable for the multicast system with high mobility than other works.

References:

[1]H. K. Aslan, “A scalable and distributed multicast security protocol using a subgroup-key hierarchy,” *Computers and Security*, vol. 23, pp. 320-329, 2004.

[2] M. Steiner, G. Tsudik and M. Waidner, “Key agreement in dynamic peer groups,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769-780, August 2000.

[3] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, 1976.

[4] I. Ingemarsson, D. Tang and C. Wong, “A conference key distribution system,” *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 714-720, September 1982.

[5] G.H. Chiou and W.T. Chen, “Secure broadcasting using the secure lock,” *IEEE Transactions on Software Engineering*, vol. 15, no. 8, pp. 929-934, August 1989.

[6] C.K. Wong, M. Gouda and S.S. Lam, “Secure group communications using key graphs,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, February 2002.

[7] M. Hiltunen and R. Schlichting, “A configurable membership service,” *IEEE Transactions on Computers*, vol. 47, no. 5, pp. 573–586, May 1998.

[8] H. Harney and C. Muckenhirn, “Group key management protocol (GKMP) architecture,” *Internet Engineering Task Force*, RFC 2094, July 1997.

[9] S. Mittra, “Iolus: A framework for scalable secure multicasting,” *Proceedings of ACM SIGCOMM’97*, Cannes, France, pp. 277–278, September 1997.

[10]M. Reiter, “Secure agreement protocols: reliable and atomic group multicast in rampart,” *Proceedings of 2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia, USA, pp. 68–80, November 1994.

[11]P. McDaniel, A. Prakash, and P. Honeyman, ”Antigone: A flexible framework for secure group communication,” *Proceedings of 8th USENIX UNIX Security Symposium*, Washington D. C., USA, pp. 99–114, August 1999.

[12]K. Birman, “The process group approach to reliable distributed computing,” *Communications of the ACM*, vol. 36, no. 12, pp. 37–53, December 1993.

[13]G. Ateniese, M. Steiner, and G. Tsudik, “New multiparty authentication services and key agreement protocols,“ *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 628-639, 2000.

[14] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, pp. 612-613, November 1976.

[15] E. Biham and A. Shamir, “Differential cryptanalysis of DES-like cryptosystems,” *Journal of Cryptology*, vol. 4, no. 1, pp. 3-72, 1991.

[16] D. W. Davies, “Some regular properties of DES,” *Advances in Cryptology Proceedings of Crypto’82*, plenum press, pp. 89-96, 1983.

[17] R. M. Davis, “The data encryption standard in perspective,” Computer Security and Data Encryption Standard, National Bureau of Standards Special Publication, February 1978.

[18] S. Miyaguchi, “The FEAL cipher family,” *Advances in Cryptology Proceedings of Crypto’90*, Berlin: Springer-Verlag, pp.

627-638, 1991.

[19] A. Shimizu and S. Miyaguchi, “Fast data encipherment algorithm FEAL,” *Advances in Cryptology Proceedings of EUROCRYPT’87*, Berlin: Springer-Verlag, pp. 267-278, 1987.