Distributed Dynamic Multicast Key Management Scheme Based on Grouped Linked List

Liling Cao, Mei Liang, Yu Zhang, and Shouqi Cao
(Corresponding author: Shouqi Cao)

Department of Engineering Science and Technology, Shanghai Ocean University
Shanghai 201306, China
Email: llcao@shou.edu.cn

(Received July 20, 2021; Revised and Accepted June 6, 2022; First Online July 3, 2022)

Abstract

Multicast technology can establish point-to-multipoint network connections among multicast group members, which can effectively solve the problems of communication transmission from single point to multi-point and is widely used in the field of internet information service. Multicast key management is the key to secure multicast communication. With the increment of multicast group members, the key technologies in existing multicast key management schemes, such as session key agreement among group members and the key update caused by dynamic changes of group members, are facing new challenges. Most of the existing multicast key management schemes rely on the logical key tree, which is not suitable for networks with many group members. In this paper, based on a hierarchical distributed mechanism, a dynamic multicast key management scheme based on a grouped linked list is proposed using timestamps, hash functions, elliptic curve cryptosystems, etc. The proposed scheme is proved secure in formal analysis based on BAN logic. Security characteristics analysis shows that the proposed scheme is suitable for multicast communication with large-scale dynamic groups with high-security performance. It can provide confidentiality, forward security, and backward security in multicast communication and resist replay attacks and conspiracy attacks. Meanwhile, the proposed scheme improves the efficiency in agreement and updates the group session key.

Keywords: Dynamic Group; Group Linked List; Key Management; Multicast

1 Introduction

Group-oriented internet information services, such as online live broadcasting, web TV, distance education, telemedicine, real-time video conferencing, etc., usually need to establish a secure channel to safely transmit data from an entity to a group of receivers through an open network. Take the electronic health social system shown in Figure 1 for instance, the patients diagnosed and treated by the same medical institution form a patient group. Communication between different groups with safely established group session key may provide patients an effective way to share treatment experience, exchange medical information and establish supportive relationships [25].

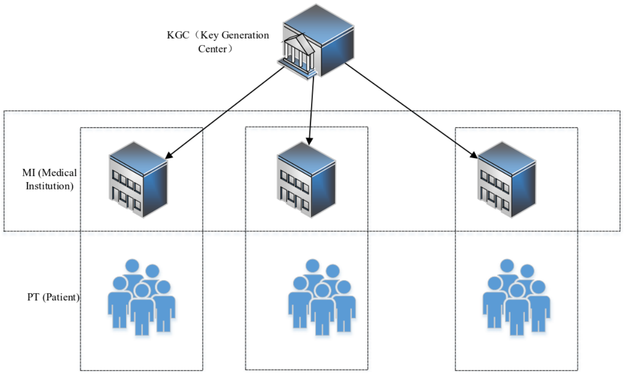


Figure 1: Electronic health social system

In order to realize such a secure group communication channel, group users need to share a group session key. Only group users who have the shared group session key can decrypt ciphertext generated by other group users. The purpose of multicast key management is to achieve efficient update of group session keys on the basis of ensuring communication security, reducding the calculation overhead of group session key update, and ensuring that only legitimate members of the group can own the group session key, and adapting to the dynamic change of membership, that is, when a user leaves or joins a group, the leaving user cannot obtain the communication data in the future group, and the newly added user cannot decrypt the communication data in the previous group. The security performance achieved by the efficient and secure multicast key management solution includes providing confidentiality, forward security, backward security, resistance to replay attacks, accomplice attacks, and providing high efficiency with small overhead in calculation, storage and network communication. The most effective method to improve the efficiency is to reduce the communication overhead. Since the scale of group members is huge, the reduction of communication overhead will greatly reduce the transmission burden of the network and effectively improve the efficiency of key management [1,12,14,27]. According to the organizational structure, the existing multicast key management solutions can be divided into three categories: centralized, decentralized, and distributed.

In the centralized multicast key management scheme, there is a group controller (GC) to manage group members. Most existing centralized multicast key management schemes are based on the Logic Key Hierarchy (LKH) structure. The use of LKH facilitates user management and reduces the calculation overhead in key update. But as the number of users increases, the depth of LKH will increase, and the calculation overhead in key update will increase. Chen Jianwei et al. proposed a multicast key management scheme using pre-distributed keys [5], which has a large security risk and high storage overhead. Chen Yanli et al. proposed a multicast key management scheme based on attribute encryption to prevent collusion attacks [6], which used combination theory to generate secret information of group members. When a member leaves or joins a group, the group controller will generate key update information based on attribute encryption. The centralized multicast key management scheme has high management efficiency and low overhead, but has the problems of poor reliability, poor scalability, and low performance during long-term operation. The entire correspondence between group members will be cut off when the the group controller GC fails. Therefore, centralized multicast key management scheme is not suitable for large-scale dynamic groups.

In a decentralized (also known as grouping) multicast key management scheme, the entire group is divided into multiple groups, and each group has a subgroup controller responsible for key management within the subgroup. Setia et al. proposed a method to update the group session key regularly [17]. This method does not consider dynamic changes such as members leaving or joining, and the group key is updated regularly. If an attacker obtains one of the group session keys, all subsequent group session keys may be compromised. The Iolus scheme [15] proposed by Mittra divides all members into several subgroups, each of which is independent of each other, which greatly reduces the cost of key updates. However, frequent encryption and decryption in the communication process will increase the computational overhead of the system, resulting in data transmission delay. Hu Yunsong et al. proposed a topology information-based multicast key management scheme [10]. The scheme constructed a logical key tree by analyzing the topology information between nodes, and optimized the hierarchical structure of the key tree to improve the efficiency of group session key update. The improvement of efficiency of the group key update also reduces energy consumption during the update process. Tan Zhigang et al. used an intra-cluster grouping method for group key management in the cluster [20]. The elliptic curve encryption group key was used for key updates, which can effectively ensure the security of group key transmission and greatly reduce energy consumption. Zhang Hui proposed a key management scheme for large-scale multicast communication [26]. The communication overhead of this scheme is significantly less than that of the LKH tree-based schemes and the Iolus scheme, and it is more suitable for large-scale secure multicast communication. The decentralized (also known as grouping) multicast key management scheme has good scalability and is suitable for dynamic groups [2,28].

There is no group controller in the distributed multicast key management scheme, and group members have the same state and status [16]. The group session key is generated by a temporary key randomly selected by each group member through negotiation. Steiner et al. proposed a multicast key management scheme Clique [18] based on the Diffie-Hellman key agreement algorithm. The scheme has good scalability, but the protocol cannot resist man-in-the-middle attacks, replay attacks, and internal node collusion attack. Xu Jianzhen et al. proposed a trust-based multicast key management scheme [24], in which a trustworthiness mechanism was established on the server node. By calculating the trustworthiness, nodes with high trustworthiness were selected as the server node to shorten the delay time of key update and improve the efficiency of key update. However, when group members leave or join frequently, the communication overhead will increase. Vijayakumar et al. proposed a single-round protocol that used the Chinese remainder theorem for group session key calculation [21], and reduced the computational complexity of group users to through the RSA encryption algorithm. Hussein et al. proposed a new scalable group distributed management method based on elliptic curve encryption [11], which ensured that the information exchange between the layers of the Internet of Things framework was not affected by deliberate attacks. Each member uses authentication information, and group user key negotiation takes only two rounds. The group members of distributed multicast key management scheme have great freedom, and all the members produce group session keys together, which improves the reliability of the key.

To sum up, most centralized multicast key management schemes are implemented based on LKH, with poor reliability and scalability. The decentralized multicast key management scheme without group controller can solve the problem of poor scalability and improve efficiency and security. However, the decentralized multicast key management scheme needs to be divided into subgroups, and the division of subgroups will inevitably affect the data transmission path. The distributed group key management scheme can precisely solve this problem [4,19,22]. In this paper, combining the advantages of decentralized and distributed multicast key management schemes, and basing on the characteristics and requirements of existing multicast applications, a distributed dynamic multicast key management scheme based on grouped linked list is proposed in this paper. Based on the characteristics of decentralized key management, the group is divided into several small subgroups, while distributed key management mechanism is adopted among all subgroups and members of subgroups to avoid this type of situation that group session keys are distributed by the group control center. The analysis of security and efficiency shows that our proposed scheme is suitable for large dynamic groups with high security performance, and can provide confidentiality, forward security, backward security, resist replay attack and collusion attack. Meanwhile, the proposed scheme improves the efficiency in agreement and update of group session key.

2 Distributed Dynamic Multicast Key Management Scheme Based on Grouped Linked List

The symbols involved in the multicast key management scheme and their meanings are shown in Table 1, and the membership structure of the multicast group in the scheme is shown in Figure 2. The members of a multicast group are composed of N subgroups, each of which forms its own member list. The first member in the list is also called the communication group's member, and N members of the communication group constitute a communication group. The communication entities in the scheme include communication group's member; g_i and subgroup's member $u_j^{(i)}$. Among them, communication group's members are also special subgroup's members.

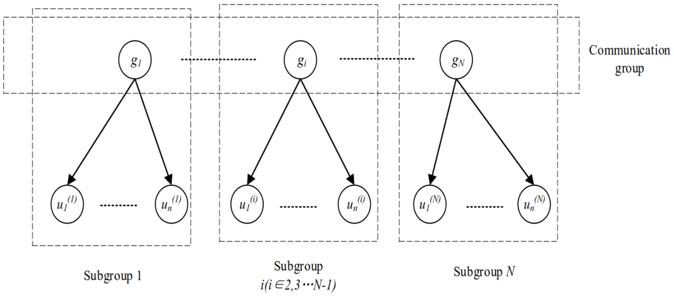


Figure 2: The proposed group member structure of the multicast key management scheme (In the picture, $g_1 = u_1^{(1)}$, $g_i = u_1^{(i)}$, $g_N = u_1^{(N)}$)

2.1 Registration Phase

Step 1: The new member provides identity information ID_j and p_j to the Key Generation Center (KGC). The new member randomly selects $z_j \in Z_n^*$ and computes the partial public key $P_j = z_jG$.

Table 1: Notations

Symbol	Define
n	Number of subgroup's members
N	Number of communication's groups
$u_j^{(i)}(1 \le j \le n, 1 \le i \le N)$	Group i subgroup's member j
$g_i(1 \le i \le N)$	Members of group i communication group
ID_j	Member's identity information
U	Group membership list
s_j	The private key of group members
p_j	The public key of group members
$SK_{i_{old}}$	The old subgroup session key of group i
SK_i	The current subgroup session key of group i
SK_i'	The updated subgroup session key for group i
KG_{old}	Old communication group session key
KG	Current communication group session key
KG'	The updated communication group session key
q	Large prime numbers
Z_n^*	$Z_n^* = [1, n-1]$
G	The base point of an elliptic curve
P	A point on an elliptic curve
x_j, y_j	The x-coordinate & y-coordinate values of a point on an elliptic curve
z_j, a_j, b_i	Random number
$\|\|$	Connection operation
$\oplus$	XOR
$h(\cdot)$	The hash function

Step 2: KGC selects the master key $s \in Z_n^*$ and a random number $t_j \in Z_n^*$, then KGC computes $T_j = t_jG$, $l_j = h(ID_j \parallel T_j \parallel P_j)$, $d_j = (t_j + sl_j) \bmod q$, and sets d_j as the partial private key. Finally, KGC sends d_j to the new member through the secure channel.

Step 3: The new member sets the private key to $s_j = z_j + d_j$ and the public key to $p_j = s_jG$.

In a large-scale dynamic group, when the number of registered members is far greater than N, KGC randomly chooses N members as the communication group's members and distributes the initial key of the communication group to them as the current communication group session key KG. The communication group members distribute the initial key of the subgroup to the corresponding subgroup's members as the key of the current subgroup $SK_i (i = 1, \cdots, N)$. The key negotiation of communication group and subgroup session keys are carried out according to the key negotiation mechanism in the following chapters.

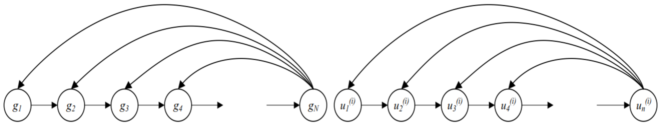
When a subgroup's member joins or leaves the group, the communication group's member is responsible for updating the subgroup's member list and key update; when the communication group's member leaves the group, the first member in the subgroup's member list becomes a new communication group's member, and the original communication group's member obtains the subgroup's member list and the communication group session key.

2.2 Key Agreement Update Mechanism

Communication group's members are responsible for generating each subgroup session key SK_i through negotiation, and update the subgroup session key and communication group session key based on grouping linked list as the subgroup's members change. Among them, the key KG of the communication group is generated by the members of the communication group g_i through mutual negotiation to realize the secure communication between subgroups.

When a new member $u_j^{(i)}$ joins the group, $u_j^{(i)}$ applies to g_i for membership, g_i updates the subgroup session key of group i, and updates the communication group session key. The subgroup session key was changed from SK_i to SK_i', and the communication group session key was changed from KG to KG'.

As shown in Figure 3, the key update of communication group and subgroup adopts the way of linked list update, and the members are arranged in order to form a linked list. When a group member (communication group member or subgroup member) needs to negotiate a group session key (communication group session key or subgroup session key), according to the order of the list of group's members, the first group member calculates the relevant information, and then sends it to the second group member, the second group member performs verification based on the obtained information, calculates the relevant information and sends it to the next group member, and so on until the last group member, verifies the information of the previous member, and calculates the group session key. The last group member calculates the relevant information based on the calculated group session key and the received information, and then sends it to the other group members in the linked list. The other group members calculate the group session key according to the data sent by the last group member.



(a) Communication group session key update linked list (b) Subgroup session keys update the linked list

Figure 3: Multicast key update mechanism

2.3 Communication Group Session Key Management

When the subgroup session key SK_i of subgroup i is updated, the communication group member g_i uses the current communication group session key KG to send multicast information to other communication group members through the multicast network, and the communication group session key is updated to KG'. In this paper, a new member of subgroup 1 joins as an example to illustrate the communication group session key update scheme.

As shown in Figure 4, when a new subgroup member $u_j^{(1)}$ makes a request to join the communication group member g_1, g_1 uses the subgroup session key SK_1 to multicast to all subgroup members, and notifies the joining of $u_j^{(1)}$. Then, the new subgroup session key SK_1' is recalculated together with the newly joined group members. Finally, the members of each communication group negotiate to generate a new communication group session key KG'.

Step 1: Initially, g_1 selects a random number b_1 and generates a timestamp t_1, then g_1 calculates $K_1 = h(b_1 \parallel t_1)$, and uses the session key SK_1' to calculate $A_1 = h(b_1 \parallel SK_1' \parallel K_1)$, $R_1 = Y_{11} = b_1P$, $TR_1 = h(KG \cdot R_1 \parallel t_1)$, $L_1 = TR_1 \oplus A_1$, $L_2 = h(A_1 \parallel R_1 \parallel t_1)$, then g_1 sends $\{L_1, L_2, t_1, R_1, Y_{11}\}$ to the next node g_2.

Step 2: $g_i (i \in \{2, 3, \cdots, N-1\})$ received a message sent by a previous communication group member.

When $i = 2$, g_2 received information $\{L_1, L_2, t_1, R_1, Y_{11}\}$;

When $i \geq 3$, g_i received the message $\{L_{2(i-1)-1}, L_{2(i-1)}, t_{i-1}, R_{i-1}, Y_{(i-1)1}, \cdots, Y_{(i-1)(i-1)}, Y_{(i-2)1}\}$.

$g_i (i \in \{2, 3, \cdots, N-1\})$ chooses a random numbers b_i, then g_i verifies if $|t_i - t_{i-1}| \leq \Delta T$, where t_i is the actual received time of the message tuple from

g_{i-1} and ΔT is the maximum transmission delay. If it fails, the session is aborted else g_i computes as follows.

$$\begin{aligned} TR'_{i-1} &= h(KG \cdot R_{i-1} \parallel t_{i-1}), \\ A'_{i-1} &= L_{2(i-1)-1} \oplus TR'_{i-1}, \\ L'_{2(i-1)} &= h(A'_{i-1} \parallel R_{i-1} \parallel t_{i-1}). \end{aligned}$$

Then g_i checks whether $L'_{2(i-1)} = L_{2(i-1)}$ holds correct. If it does not hold correct, the session is aborted else g_i calculates as follows. $K_i = h(b_i \parallel t_i)$, $A_i = h(b_i \parallel SK_i \parallel K_i)$,$R_i = b_iP$:

When $i = 2$, $Y_{21} = b_2Y_{11} = b_2b_1P$; $Y_{22} = b_2P$.

When $i \geq 3$, $Y_{i1} = b_iY_{(i-1)1} = b_i \cdots b_2b_1P$; $Y_{i2} = b_iY_{(i-1)2} = b_i \cdots b_3b_2P$; $\cdots$; $Y_{i(i-1)} = b_iY_{(i-1)(i-1)} = b_ib_{i-1}b_{i-3} \cdots b_1P$; $Y_{ii} = b_iY_{(i-2)1} = b_ib_{i-2} \cdots b_1P$, then g_i calculates $TR_i = h(KG \cdot R_i \parallel t_i)$, $L_{2i-1} = TR_i \oplus A_i$, $L_{2i} = h(A_i||R_i||t_i)$.

Finally, g_i sends $\{L_{2i-1}, L_{2i}, t_i, R_i, Y_{i1}, \cdots, Y_{ii}, Y_{(i-1)1}\}$ to the next node g_{i+1}.

Step 3: Upon receiving the message tuple $\{L_{2(N-1)-1}$, $L_{2(N-1)}$, t_{N-1}, R_{N-1}, $Y_{(N-1)1}$, $\cdots$, $Y_{(N-1)(N-1)}$, $Y_{(N-2)1}\}$, $g_N(N \geq 3)$ selects a random number b_N, g_N verifies if $|t_N - t_{N-1}| \leq \Delta T$, where t_N is the actual received time of the message tuple from g_{N-1} and ΔT is the maximum transmission delay. If it fails, the session is aborted else $g_N(N \geq 3)$ calculates as follows.

$$\begin{aligned} TR'_{N-1} &= h(KG \cdot R_{N-1} \parallel t_{N-1}), \\ A'_{N-1} &= L_{2(N-1)-1} \oplus TR'_{N-1}, \\ L'_{2(N-1)} &= h(A'_{N-1} \parallel R_{N-1} \parallel t_{N-1}). \end{aligned}$$

Then g_N checks whether $L'_{2(N-1)} = L_{2(N-1)}$ holds correct. If it does not hold correct, the session is aborted else g_N calculates as follows.

$$\begin{aligned} Y_{N1} &= b_NY_{(N-1)1} = b_N \cdots b_2b_1P, \\ Y_{N2} &= b_NY_{(N-1)2} = b_N \cdots b_3b_2P, \\ \vdots &= \vdots \\ Y_{N(N-1)} &= b_NY_{(N-1)(N-1)} = b_Nb_{N-1}b_{N-3} \cdots b_1P, \\ Y_{NN} &= b_NY_{(N-2)1} = b_Nb_{N-2} \cdots b_1P, \\ KG' &= h(KG \cdot Y_{N1}) = h(KG \cdot b_N \cdots b_2b_1P). \end{aligned}$$

Finally, g_N sends Y_{NN}, $Y_{N(N-1)}$, $\cdots$, Y_{N3}, Y_{N2} to g_{N-1}, g_{N-2}, $\cdots$, g_2, g_1, respectively.

Step 4: Upon receiving the message Y_{N2}, Y_{N3}, $\cdots$, $Y_{N(N-1)}$, Y_{NN}, $g_i(i \in \{1, 2, \cdots, N-1\})$ calculates $KG \cdot b_1Y_{N2}$, $KG \cdot b_2Y_{N3}$, $\cdots$, $KG \cdot b_{N-2}Y_{N(N-1)}$, $KG \cdot b_{N-1}Y_{NN}$ respectively, and gets the communication group session key is $KG' = h(KG \cdot b_N \cdots b_2b_1P)$.

2.4 Subgroup Session Key Management

The subgroup session key management scheme realizes the key negotiation and update among n subgroup members within the subgroup. First, initialize the subgroup i member list $U = \{u_1^{(i)}, u_2^{(i)}, \cdots u_n^{(i)}\}$, and the subgroup member list linked list remains unchanged and is shared by all subgroup members. When a member changes (for example, the departure of an old subgroup member or the addition of a new member), the list of subgroup members also changes. Each member has a private key s_j, a public key p_j, and $p_j = s_jG$. The public key is shared by all members. When a new member applies to a communication group's member g_i to join the group, the communication group's member g_i of the subgroup sends the information to all the subgroup members over the multicast network.

As shown in Figure 5, the entire subgroup session key agreement protocol is divided into three phases: the subgroup establishment and subgroup session key agreement phase, the new member joining phase, and the old member leaving phase.

2.4.1 Subgroup Establishment and Subgroup Session Key Agreement Phase

Step 1: Initially, $u_1^{(i)}$ chooses a random number a_1 and generates a timestamp T_1, then $u_1^{(i)}$ calculates $k_{1,2} = s1p2$, $k_{1,2}$ is a point on the elliptic curve, and its coordinates are $(x_{1,2}, y_{1,2})$, $u_1^{(i)}$ calculates $t_{1,2} = h(x_{1,2} \oplus y_{1,2})$, $B_1 = h(a_1||ID_1 \parallel t_{1,2})$, $g_1 = a_1P$, $X_{11} = a_1P$, $SR_1 = h(g_1 \cdot SK_i \cdot P \parallel T_1)$, $M_1 = SR_1 \oplus B_1$,$M2 = h(B_1||g_1||T_1)$, then $u_1^{(i)}$sends$\{M_1, M_2, T_1, g_1, X_{11}\}$ to the next node g_2.

Step 2: $u_j^{(i)}(j \in \{2, 3, \cdots, n-1\})$ received the message sent by the previous member.

When $j = 2$, $u_2^{(i)}$ received information $\{L_1, L_2, t_1, R_1, Y_{11}\}$;

When $j \geq 3$, $u_j^{(i)}$ received information $\{SR_{j-1}$, $M_{2(j-1)-1}$, $M_{2(j-1)}$, T_{j-1}, g_{j-1}, $X_{(j-1)1}$, $\cdots$, $X_{(j-1)(j-1)}$, $X_{(j-2)1}\}$.

$u_j^{(i)}(j \in \{2, 3, \cdots, n-1\})$ chooses a random number a_j, then $u_j^{(i)}$ verifies if $|T_j - T_{j-1}| \leq \Delta T$, where T_j is the actual received time of the message tuple from $u_{j-1}^{(i)}$ and ΔT is the maximum transmission delay. If it fails, the session is aborted else $u_j^{(i)}$ computes as follows.

$$\begin{aligned} SR'_{j-1} &= h(g_{j-1} \cdot SK_i \cdot P \parallel T_{j-1}), \\ B'_{j-1} &= M_{2(j-1)-1} \oplus SR'_{j-1}, \\ M'_{2(j-1)} &= h(B'_{j-1}||g_{j-1}||T_{j-1}). \end{aligned}$$

Then $u_j^{(i)}$ checks whether $M'_{2(j-1)} = M_{2(j-1)}$ holds correct. If it does not hold correct, the session is

g_1 g_2 ...... g_N

$$A_1 = h(b_1 \| SK'_1 \| K_1)$$
$$R_1 = Y_{11} = b_1 P$$
$$TR_1 = h(KG \cdot R_1 \| t_1)$$
$$L_1 = TR_1 \oplus A_1$$
$$L_2 = h(A_1 \| R_1 \| t_1)$$

$$\xrightarrow{\{L_1, L_2, t_1, R_1, Y_{11}\}} \quad Generate b_2 \ \& \ t_2$$

$$|t_2 - t_1| \le \Delta T$$
$$TR'_1 = h(KG \cdot R_1 \| t_1)$$
$$A'_1 = L_1 \oplus TR'_1$$
$$L'_2 = h(A'_1 \| R_1 \| t_1)$$
$$If \ L'_2 = L_2, continue$$
$$K_2 = h(b_2 \| t_2)$$
$$A_2 = h(b_2 \| SK_2 \| K_2)$$
$$R_2 = b_2 P$$
$$Y_{21} = b_2 Y_{11} = b_2 b_1 P$$
$$Y_{22} = b_2 P$$
$$TR_2 = h(KG \cdot R_2 \| t_2)$$
$$L_3 = TR_2 \oplus A_2$$
$$L_4 = h(A_2 \| R_2 \| t_2)$$

$$\xrightarrow{\{L_3, L_4, t_2, R_2, Y_{21}, Y_{22}, Y_{11}\}}$$

....

$$\xrightarrow{\{L_{2(N-1)-1}, L_{2(N-1)}, t_{N-1}, R_{N-1}, Y_{(N-1)1}, \ldots\ldots, Y_{(N-1)(N-1)}, Y_{(N-2)1}\}} \quad Generate b_N \ \& \ t_N$$

$$|t_N - t_{N-1}| \le \Delta T$$
$$TR'_{N-1} = h(KG \cdot R_{N-1} \| t_{N-1})$$
$$A'_{N-1} = L_{2(N-1)-1} \oplus TR'_{N-1}$$
$$L'_{2(N-1)} = h(A'_{N-1} \| R_{N-1} \| t_{N-1})$$
$$If \ L'_{2(N-1)} = L_{2(N-1)}, continue$$
$$A_N = h(b_N \| SK_N \| K_N)$$
$$R_N = b_N P$$
$$Y_{N1} = b_N Y_{(N-1)1} = b_N .. b_2 b_1 P$$
$$Y_{N2} = b_N Y_{(N-1)2} = b_N .. b_3 b_2 P$$

....

$$Y_{N(N-1)} = b_N Y_{(N-1)(N-1)} = b_N b_{N-1} b_{N-3} \ldots b_1 P$$
$$Y_{NN} = b_N Y_{(N-2)1} = b_N b_{N-2} .. b_1 P$$
$$KG' = h(KG \cdot b_N .. b_2 b_1 P)$$

$$KG' = h(KG \cdot b_{N-1} Y_{NN}) = h(KG \cdot b_N \ldots b_2 b_1 P) \xleftarrow{\{Y_{NN}\}}$$
$$KG' = h(KG \cdot b_{N-2} Y_{N(N-1)}) = h(KG \cdot b_N \ldots b_2 b_1 P) \xleftarrow{\{Y_{N(N-1)}\}}$$

....

$$KG' = h(KG \cdot b_1 Y_{N2}) = h(KG \cdot b_N .. b_2 b_1 P) \xleftarrow{\{Y_{N2}\}}$$

Figure 4: Communication group key agreement

aborted else $u_j^{(i)}$ calculates as follows.

$$\begin{aligned} k_{j,(j+1)} &= s_j p_{(j+1)}, \\ t_{j,(j+1)} &= h(x_{j,(j+1)} \oplus y_{j,(j+1)}), \\ B_j &= h(a_j \parallel ID_j \parallel t_{j,(j+1)}). \end{aligned}$$

When $j = 2$, $X_{21} = a_2 a_1 P$; $X_{22} = a_2 P$.

When $j \geq 3$, $X_{j1} = a_j X_{(j-1)1} = a_j \cdots a_2 a_1 P$; $X_{j2} = a_j X_{(j-1)2} = a_j \cdots a3 a_2 P$; $\cdots$, $X_{j(j-1)} = a_j X_{(j-1)(j-1)} = a_j a_{j-1} a_{j-3} \cdots a_1 P$; $X_{jj} = a_j X_{(j-2)1} = a_j a_{j-2} \cdots a_1 P$.

Then $u_j^{(i)}$ computes $SR_j = h(g_j \cdot SK_i \cdot P \parallel T_j)$, $M_{2j-1} = SR_j \oplus B_j$, $M_{2j} = h(B_j || g_j || T_j)$.

Finally, $u_j^{(i)}$ sends $\{M_{2j-1}, M_{2j}, T_j, g_j, X_{j1}, \cdots, X_{jj}, X_{(j-1)1}\}$ to the next node $u_{j+1}^{(i)}$.

Step 3: Upon receiving the message tuple $\{M_{2(n-1)-1}$, $M_{2(n-1)}$, T_{n-1}, g_{n-1}, $X_{(n-1)1}$, $\cdots$, $X_{(n-1)(n-1)}$, $X_{(n-2)1}\}$, $u_n^{(i)}(n \geq 3)$ selects a random numbera_n, then verifies if $|t_n - t_{n-1}| \leq \Delta T$, where T_n is the actual received time of the message tuple from u_{n-1}^i and ΔT is the maximum transmission delay. If it fails, the session is aborted else u_n^i calculates as follows.

$$\begin{aligned} SR'_{n-1} &= h(g_{n-1} \cdot SK_i \cdot P \parallel T_{n-1}), \\ B'_{n-1} &= M_{2(n-1)-1} \oplus SR'_{n-1}, \\ M'_{2(n-1)} &= h(B'_{n-1} || g_{n-1} || T_{n-1}). \end{aligned}$$

Then u_n^i checks whether $M'_{2(n-1)} = M_{2(n-1)}$ holds correct. If it does not hold correct, the session is aborted else u_n^i calculates as follows.

$$\begin{aligned} X_{n1} &= a_n X_{(n-1)1} = a_n \cdots a_2 a_1 P, \\ X_{n2} &= a_n X_{(n-1)2} = a_n \cdots a3 a_2 P, \\ \vdots &= \vdots \\ X_{n(n-1)} &= a_n X_{(n-1)(n-1)} = a_n a_{n-1} a_{n-3} \cdots a_1 P, \\ X_{nn} &= a_n X_{(n-2)1} = a_n a_{n-2} \cdots a_1 P, \\ SK'_i &= h(SK_i \cdot P \parallel X_{n1}) \\ &= h(SK_i \cdot P \parallel a_n \cdots a_2 a_1 P). \end{aligned}$$

Finally, u_n^i sends X_{nn}, $X_{n(n-1)}$, $\cdots$, X_{n3}, X_{n2} to u_{n-1}, u_{n-2}, $\cdots$, u_2, u_1, respectively.

Step 4: Upon receiving the message X_{n2}, X_{n3}, $\cdots$, $X_{n(n-1)}$, X_{nn}, $u_j^{(i)}(j \in \{1, 2, \cdots, n-1\})$ calculates $a_1 X_{n2}$, $a_2 X_{n3}$, $\cdots$, $a_{n-2} X_{n(n-1)}$, $a_{n-1} X_{nn}$, respectively, and gets the subgroup session key is $SK'_i = h(SK_i \cdot P \parallel a_n \cdots a_2 a_1 P)$.

2.4.2 New Member Joining Phase

1) When a new member joins a subgroup communication, according to the member's personal information, g_i will assign the new member to the corresponding position of the group and update the group's member list $U = \{u_1^{(i)}, u_2^{(i)}, \cdots, u_n^{(i)}, u_{n+1}^{(i)}\}$, then g_i distributes the subgroup session key $SK_i \cdot P$ to the new members.

2) At this time, the key negotiation needs to be performed again, and the steps are the same as $u_j^{(i)}$ Step 1 to Step 4 of the subgroup establishment and subgroup session key negotiation phase.

2.4.3 The Old Members Leave Phase

1) When member $u_j^{(i)}$ leaves the group, the group member list $U = \{u_1^{(i)}, u_2^{(i)}, \cdots, u_{n-1}^{(i)}\}$ will be updated again according to the position of the leaving member, that is, the leaving member will be deleted; if the member who has left is a communication group member, the second member in the list becomes the new communication group member.

2) At this time, the key agreement needs to be renegotiated. The steps of key agreement are the same as Step 1 to Step 4 of the subgroup establishment and subgroup session key negotiation phases.

3 Performance Analysis

3.1 Efficiency Analysis

The dynamic multicast scheme based on the grouped linked list proposed in this paper combines the advantages of decentralized and distributed multicast key management schemes. Comparing with the LKH-based scheme, the Iolus scheme and several key management schemes based on elliptic curve cryptosystems, the overall efficiency is higher. The comparison results are shown in Table 2, where the parameters used in the table are as follows:

n**:** The total number of members of the entire group;

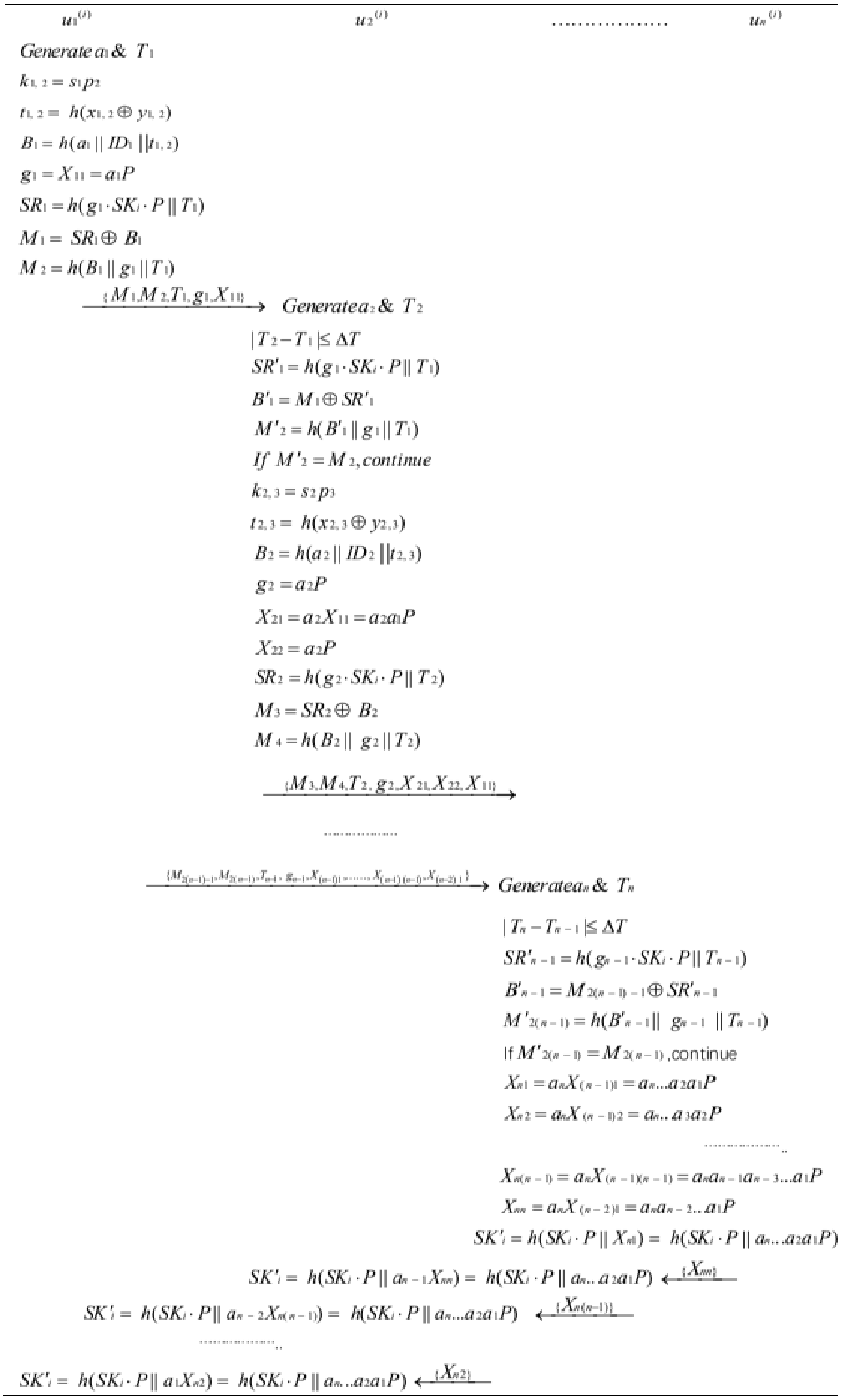
N**:** The number of subgroup;

m**:** The number of members within each subgroup.

Among them, the total number of members of the entire group $n =$ The number of subgroups $N \times$ The number of members within each subgroup m.

In this scheme, the key storage capacity of the members of the subgroup is *1*, and the key storage capacity of the members of the communication group is *2*. A key negotiation is conducted in the subgroup, the number of communication rounds for each user is *1*, and the number of calculation times is *2*. Compared with literature [7, 8, 23], the number of communication rounds of users is reduced, which has a higher efficiency. Compared with literature [9], it can adapt to the dynamic changes of members.

Compared with the scheme based on LKH, this scheme reduces the number of key stores in member nodes. In



$u_1^{(i)}$ $u_2^{(i)}$ ……………… $u_n^{(i)}$

$Generate\ a_1 \,\&\, T_1$

$k_{1,2} = s_1 p_2$

$t_{1,2} = h(x_{1,2} \oplus y_{1,2})$

$B_1 = h(a_1 \| ID_1 \| t_{1,2})$

$g_1 = X_{11} = a_1 P$

$SR_1 = h(g_1 \cdot SK_i \cdot P \| T_1)$

$M_1 = SR_1 \oplus B_1$

$M_2 = h(B_1 \| g_1 \| T_1)$

$\xrightarrow{\{M_1, M_2, T_1, g_1, X_{11}\}}$ $Generate\ a_2 \,\&\, T_2$

$|T_2 - T_1| \leq \Delta T$

$SR'_1 = h(g_1 \cdot SK_i \cdot P \| T_1)$

$B'_1 = M_1 \oplus SR'_1$

$M'_2 = h(B'_1 \| g_1 \| T_1)$

$If\ M'_2 = M_2, continue$

$k_{2,3} = s_2 p_3$

$t_{2,3} = h(x_{2,3} \oplus y_{2,3})$

$B_2 = h(a_2 \| ID_2 \| t_{2,3})$

$g_2 = a_2 P$

$X_{21} = a_2 X_{11} = a_2 a_1 P$

$X_{22} = a_2 P$

$SR_2 = h(g_2 \cdot SK_i \cdot P \| T_2)$

$M_3 = SR_2 \oplus B_2$

$M_4 = h(B_2 \| g_2 \| T_2)$

$\xrightarrow{\{M_3, M_4, T_2, g_2, X_{21}, X_{22}, X_{11}\}}$

………………

$\xrightarrow{\{M_{2(n-1)-1}, M_{2(n-1)}, T_{n-1}, g_{n-1}, X_{(n-1)1}, \ldots, X_{(n-1)(n-1)}, X_{(n-2)1}\}}$ $Generate\ a_n \,\&\, T_n$

$|T_n - T_{n-1}| \leq \Delta T$

$SR'_{n-1} = h(g_{n-1} \cdot SK_i \cdot P \| T_{n-1})$

$B'_{n-1} = M_{2(n-1)-1} \oplus SR'_{n-1}$

$M'_{2(n-1)} = h(B'_{n-1} \| g_{n-1} \| T_{n-1})$

If $M'_{2(n-1)} = M_{2(n-1)}$,continue

$X_{n1} = a_n X_{(n-1)1} = a_n \ldots a_2 a_1 P$

$X_{n2} = a_n X_{(n-1)2} = a_n \ldots a_3 a_2 P$

………………..

$X_{n(n-1)} = a_n X_{(n-1)(n-1)} = a_n a_{n-1} a_{n-3} \ldots a_1 P$

$X_{nn} = a_n X_{(n-2)1} = a_n a_{n-2} \ldots a_1 P$

$SK'_i = h(SK_i \cdot P \| X_{n1}) = h(SK_i \cdot P \| a_n \ldots a_2 a_1 P)$

$SK'_i = h(SK_i \cdot P \| a_{n-1} X_{nn}) = h(SK_i \cdot P \| a_n \ldots a_2 a_1 P) \xleftarrow{\{X_{nn}\}}$

$SK'_i = h(SK_i \cdot P \| a_{n-2} X_{n(n-1)}) = h(SK_i \cdot P \| a_n \ldots a_2 a_1 P) \xleftarrow{\{X_{n(n-1)}\}}$

………………..

$SK'_i = h(SK_i \cdot P \| a_1 X_{n2}) = h(SK_i \cdot P \| a_n \ldots a_2 a_1 P) \xleftarrow{\{X_{n2}\}}$

Figure 5: Subgroup key agreement

Table 2: Efficiency comparison of key management solutions

Scheme	Key storage		Communication overhead		Number of member communication rounds	Number of member counts	Adapt to dynamic changes in members
	Server	Members of the node	Join	Leave			
LKH	$O(n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	1	2	yes
Iolus [20]	$O(m)$	$O(1)$	$O(1)$	$O(m)$	/	/	yes
Choi [7]	/	$O(n)$	$O(n)$	$O(n)$	2	3	no
Gorantla [9]	/	$O(n)$	$O(n)$	$O(n)$	1	2	no
Xie [23]	/	$O(n/3)$	$O(n)$	$O(n)$	$2n/3$, $(n{=}3k, n{\in}\ z)$	More	no
Gao [8]	$O(n)$	$O(3)$	$O(n)$	$O(n)$	2	3	yes
Ours	$O(2)$	$O(1)$	$O(m)$	$O(m)$	1	2	yes

the Iolus scheme, because the load of decryption/re-encryption is borne by the local group security intermediate node (GSI), it is easy to form a bottleneck, and in this scheme, there is no such problem, so the efficiency is higher.

3.2 Security Analysis

3.2.1 Formal Analysis of BAN Logic

BAN logic formal analysis method is used to prove the security of the scheme. Since communication group session key management and subgroup session key management have the same structure, this section takes subgroup session key management as an example to carry out security proof. The analysis of the subgroup session key management scheme using BAN logic consists of the following four steps:

1) Security objectives of the scheme

G1: $u_j^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

G2: $u_n^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

G3: $u_j^{(i)} \mid\equiv u_n^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

G4: $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

2) Ideal model of the scheme

Msg1: $u_j^{(i)} \to u_n^{(i)} : (a_j)_{SK_{i_{old}}\cdot P}$

Msg2: $u_n^{(i)} \to u_j^{(i)} : (a_n, \cdots, a_{j+1}, a_j, \cdots, a_1)_{SK_{i_{old}}\cdot P}$

Msg3: $u_j^{(i)} \to g_i : (a_n, \cdots, a_3, a_2, a_1, g_i \overset{SK_i}{\longleftrightarrow} u_j^{(i)})_{SK_{i_{old}}\cdot P}$

Msg4: $g_i \to u_j^{(i)} : (a_n, \cdots, a_3, a_2, a_1, g_i \overset{SK_i}{\longleftrightarrow} u_j^{(i)})_{SK_{i_{old}}\cdot P}$

3) Initialize the hypothesis

P1: $u_j^{(i)} \mid\equiv u_j^{(i)} \overset{h(SK_i\cdot P\|a_n\cdots a_2a_1P)}{\longleftrightarrow} u_n^{(i)}$

P2: $u_n^{(i)} \mid\equiv u_j^{(i)} \overset{h(SK_i\cdot P\|a_n\cdots a_2a_1P)}{\longleftrightarrow} u_n^{(i)}$

P3: $u_j^{(i)} \mid\equiv u_n^{(i)} \mid\equiv u_j^{(i)} \overset{h(SK_i\cdot P\|a_n\cdots a_2a_1P)}{\longleftrightarrow} u_n^{(i)}$

P4: $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\equiv u_j^{(i)} \overset{h(SK_i\cdot P\|a_n\cdots a_2a_1P)}{\longleftrightarrow} u_n^{(i)}$

P5: $u_j^{(i)} \mid\equiv \#(a_k), (1 \le k \le n, k \ne j)$

P6: $u_n^{(i)} \mid\equiv \#(a_j), (1 \le j < n)$

P7: $u_j^{(i)} \mid\equiv u_n^{(i)} \mid\Rightarrow u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

P8: $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\Rightarrow u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

4) Based on the above initial state and BAN logic reasoning rules, it can be proved that the scheme can achieve the security objectives proposed above.

From Msg1, we get

A1: $u_n^{(i)} \triangleleft \langle a_j\rangle_{SK_{i_{old}}\cdot P}, (1 \le j < n)$

A2: According to P2, A2 is obtained by applying the message meaning rule.

A2: $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\sim \langle a_j\rangle_{SK_{i_{old}}\cdot P}$

A3: $u_n^{(i)} \mid\equiv \#\langle a_j\rangle_{SK_{i_{old}}\cdot P}$ can be obtained from P6.

A4: Based on A2 and A3, $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\equiv \langle a_j\rangle_{SK_{i_{old}}\cdot P}$ can be obtained according to the random number verification rule. And then according to Msg3, we get

A4: $g_i \triangleleft (a_n, \cdots, a_3, a_2, a_1, g_i \overset{SK_i}{\longleftrightarrow} u_j^{(i)})_{SK_{i_{old}}\cdot P}$

A5: Then, according to P2, we get

A5: $g_i \mid\equiv u_j^{(i)} \mid\sim (a_n, \cdots, a_3, a_2, a_1)_{SK_{i_{old}}\cdot P}$

A6: $g_i \mid\equiv \#(a_n, \cdots, a_3, a_2, a_1)_{SK_{i_{old}}\cdot P}$ can be obtained from P6.

A7: Based on A5 and A6, $g_i \mid\equiv u_j^i \mid\equiv (a_n, \cdots, a_3, a_2, a_1)_{SK_{i_{old}}\cdot P}$ can be obtained according to the random number verification rule.

According to A6 and P4, $SK_i = h(SK_{i_{old}\cdot P} \parallel a_n \cdots a_2 a_1 P)$.

We can get

A7(G4): $u_n^{(i)} \mid\equiv u_j^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$

A8: Based on A7 and P8,

A8(G2): $u_n^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)}$ can be obtained according to the arbitration rules.

A9: According to Msg2, we get

A9: $u_j^{(i)} \triangleleft (an, \cdots, a_{j+1}, a_j, \cdots, a_1, u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)})_{SK_{i_{old}}\cdot P}$.

A10: Then according to P1,

A10: $u_j^{(i)} \mid\equiv u_n^{(i)} \mid\sim (a_n, \cdots, a_{j+1}, a_j, \cdots, a_1, u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)})_{SK_{i_{old}}\cdot P}$ can be obtained by applying the message meaning rule.

A11: We get

A11: $u_j^{(i)} \mid\equiv \#(a_n, \cdots, a_{j+1}, a_j, \cdots, a_1, u_j^{(i)} \overset{SK_i}{\longleftrightarrow} u_n^{(i)})_{SK_{i_{old}}\cdot P}$ from P5.

A12: Based on A10 and A11,

A12(G3): $u_j^{(i)} \mid\equiv u_n^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longrightarrow} u_n^{(i)}$ can be obtained according to the random number verification rule.

Based on A12 and P7, G1: $u_j^{(i)} \mid\equiv u_j^{(i)} \overset{SK_i}{\longrightarrow} u_n^{(i)}$ is obtained according to the arbitration rules.

To sum up, all protocol targets G1, G2, G3 and G4 can be pushed down through formal analysis of BAN logic. Therefore, the scheme can achieve secure group session key negotiation and secure group session key establishment for group members, and finally achieve secure communication.

3.2.2 Security Characteristics Analysis

Security is the key element for the evaluation of multicast key management scheme in multicast communication, which mainly includes confidentiality, forward security, backward security, resistance to collusion attack and resistance to replay attack [3,13].

1) Confidentiality

According to Figures 3 and 4, it can be seen that any member who does not belong to the subgroup member list cannot calculate the updated subgroup session key SK_i' and the updated communication group session key KG'. The subgroup members calculate the relevant values in turn according to the list and send them to the next group member, even if the attacker intercepts the information or pretends to be a legitimate group member, because they cannot know the current subgroup session key SK_i, the current communication group session key KG, the random number a_j of subgroup members and the random number b_i of communication group members cannot calculate the corresponding $SK_i' = h(SK_i \cdot P \parallel a_n \cdots a_2 a_1 P)$, $KG' = h(KG \cdot b_n \cdots b_2 b_1 P)$, which means that the attacker can neither pretend to be a member of the subgroup nor participate in the key agreement process as a member of the communication group, nor calculate SK_i' and KG' based on the stolen information. So this scheme can guarantee the confidentiality of communication.

2) Forward safety

The member $u_j^{(i)}$'s departure process of Group i is as follows:

a. Subgroup members of group i encrypt secure communication according to subgroup session key SK_i, subgroup member $u_j^{(i)}$ exits the group, group member g_i multicast the subgroup members of group i, and removes the members who leave the subgroup from the member list;

b. The remaining subgroup members of Group i will re-negotiate the key. According to the confidentiality, any member not in the subgroup member list cannot calculate the subgroup session key SK_i' and the communication group session key KG', so the group members who have left cannot continue to participate in the key negotiation;

As can be seen from the above, the member who have left cannot use the previous subgroup session key to decode the communication content, which ensures the forward security of the information.

3) Backward Security

The process for member $u_{n+1}^{(i)}$ to join Group i is as follows:

a. The key of the current communication group is $KG = h(KG_{old} \cdot b_n \cdots b_2 b_1 P)$, and the key of the current subgroup of group i is $SK_i = h(SK_{i_{old}\cdot P} \parallel a_n \cdots a_2 a_1 P)$. When the member $u_{n+1}^{(i)}$ applies to join the i^{th} group, g_i updates the member list, the order of the linked list members changes, and g_i distributes $SK_i \cdot P$ to new members;

b. Members of the subgroup conduct key negotiation again after calculation in linked list order. The negotiation generates a new subgroup session key $SK_i' = h(SK_i \cdot P \parallel a_{n+1} a_n \cdots a_2 a_1 P)$, and the communication group members negotiate generates a new communication group session key $KG' = h(KG \cdot b_n \cdots b_2 b_1 P)$.

Based on the discrete logarithm problem of elliptic curve cryptosystem, u_{n+1} cannot use $SK_i \cdot P$ to calculate SK_i, so the historical information before u_{n+1} joins the i^{th} group is still confidential to u_{n+1}, so the new member can only obtain the key SK_i' of the new subgroup after joining the group, and the previous subgroup communication cannot be broken, which ensures the backward security of the information.

4) Anti-accomplice Cracking
For a subgroup i with n members, the left subgroup members are not in the member list, and the private key of the subgroup members and the generated random number $a_j(1 \leq j \leq n)$ cannot be learned. The calculation of the subgroup session key negotiation involves the new B_j, while the calculation of B_j involves the private key of the group members and the random number $a_j(1 \leq j \leq n)$. After each member leaves the subgroup, the group (communication group and subgroup) session key negotiation needs to be redone, and the previous subgroup session key and communication group session key are no longer used. Even if the leaving members can share information, the new $X_{jk}(1 \leq j \leq n, 1 \leq k \leq j)$ and $Y_{ij}(1 \leq i \leq n, 1 \leq j \leq i)$ cannot be calculated, so multiple leaving group members (multiple attackers) cannot obtain new group keys (communication group and subgroup) through cooperation, which can resist collusion attack.

5) Resist Replay Attacks

a. Communication group key negotiation process
Members of the communication group participate in the calculation together and obtain the key of the communication group through negotiation. Assume that the attacker intercepts the transmitted data and participates in the negotiation process of the key disguised as a legitimate member through replay attack. The detailed process is as follows:

i. g_i randomly chooses b_i and a timestamp t_i, then computes $K_i = h(b_i \parallel ti)$, $A_i = h(b_i \parallel SK_i \parallel K_i)$, $R_i = b_i P$, $Y_{ij}(1 \leq j \leq i)$, $TR_i = h(KG \cdot R_i \parallel t_i)$, $L_{2i-1} = TR_i \oplus A_i$, $L_{2i} = h(A_i || R_i || t_i)$, and sends $\{L_{2i-1}, L_{2i}, t_i, R_i, Y_{ij}(1 \leq j \leq i)\}$ to g_{i+1};

ii. g_{i+1} randomly chooses b_{i+1},then g_{i+1} verifies if $|t_{i+1} - t_i| \leq \Delta T$, where t_{i+1} is the actual received time of the message tuple from g_i and ΔT is the maximum transmission delay. If it fails, the session is aborted else g_{i+1} computes TR'_i, A'_i, L'_{2i}. Then g_{i+1} checks whether computed $L'_{2i} = L_{2i}$ hold correct. If it does not hold correct, the session is aborted else g_{i+1} calculates the communication group session key KG'.

$$\begin{aligned} TR'_i &= h(KG \cdot R_i \parallel t_i) \\ A'_i &= L_{2i-1} \oplus TR'_i \\ L'_{2i} &= h(A'_i || R_i || t_i) \\ KG' &= h(KG \cdot b_{i+1} Y_{n(i+2)}) \\ &= h(KG \cdot b_n \cdots b_2 b_1 P). \end{aligned}$$

iii. The attacker replays the message $\{L_{2i-1}, L_{2i}, t_i, R_i, Y_{ij}(1 \leq j \leq i)\}$ to g_{i+1};

iv. g_{i+1} randomly chooses a random number b'_{i+1} and a timestamp t'_{i+1}. At this time, $|t'_{i+1} - t_i| \leq \Delta T$ is not valid and the session is aborted; if the attacker changes the timestamp, he/she sends the message $\{L_{2i-1}, L_{2i}, t'_i, R_i, Y_{ij}(1 \leq j \leq i)\}$ to g_{i+1}, $|t'_{i+1} - t'_i| \leq \Delta T$ holds, calculates TR'^{*}_i, A'^{*}_i, L'^{*}_{2i}.

$$\begin{aligned} TR'^{*}_i &= h(KG \cdot R_i \parallel t'_i) \\ A'^{*}_i &= L_{2i-1} \oplus TR'^{*}_i \\ L'^{*}_{2i} &= h(A'^{*}_i || R_i || t'_i). \end{aligned}$$

From the above, $L'^{*}_{2i} \neq L_{2i}$can be known, and the attacker can't know KG, the session is aborted. Therefore, the attacker cannot participate in the process of negotiating the key of the communication group by replaying the information.

b. Subgroup session key negotiation process
Assume that the attacker intercepts the transmitted data and participates in the key negotiation process disguised as a legitimate member through the replay attack. The detailed process is as follows:

i. $u_j^{(i)}$ randomly chooses a_j and a timestampT_j, and computes $k_{j,j+1} = s_j p_{j+1}$, $t_{j,j+1} = h(x_{j,j+1} \oplus y_{j,j+1})$, $B_j = h(a_j || ID_j \parallel t_{j,j+1})$, $g_j = a_j P$, $X_{jk}(1 \leq k \leq j)$, $SR_j = h(g_j \cdot SK_i \cdot P \parallel T_j)$, $M_{2j-1} = SR_j \oplus B_j$, $M_{2j} = h(B_j || g_j || T_j)$, then sends $\{SR_j, M_{2j-1}, M_{2j}, T_j, g_j, X_{jk}(1 \leq k \leq j)\}$ to $u_{j+1}^{(i)}$;

ii. $u_{j+1}^{(i)}$ randomly chooses a_{j+1}, then verifies if $|T_{j+1} - T_j| \leq \Delta T$, where T_{j+1} is the actual received time of the message tuple from $u_j^{(i)}$ and ΔT is the maximum transmission delay. If it fails, the session is aborted else $u_{j+1}^{(i)}$ calculatesSR'_j, B'_j, M'_{2j}.
Then $u_{j+1}^{(i)}$ checks whether computed $M'_{2j} = M_{2j}$ hold correct. If it does not hold correct, the session is aborted else $u_{j+1}^{(i)}$ calculates the subgroup session key SK'_i.

$$\begin{aligned} SR'_j &= h(g_j \cdot SK_i \cdot P \parallel T_j) \\ B'_j &= M_{2j-1} \oplus SR'_j \\ M'_{2j} &= h(B'_j || g_j || T_j) \\ SK'_i &= h(SK_i \cdot P \parallel a_{j+1} X_{n(j+2)}) \\ &= h(SK_i \cdot P \parallel a_n \cdots a_2 a_1 P). \end{aligned}$$

iii. The attacker replays the message $\{M_{2j-1}, M_{2j}, T_j, g_j, X_{jk}(1 \leq k \leq j)\}$ to $u_{j+1}^{(i)}$;

iv. $u_{j+1}^{(i)}$ randomly chooses a'_{j+1} and a timestamp T'_{j+1}. At this time, $|T'_{j+1} -$

$T_j| \leq \Delta T$ is not valid and the session is aborted; if the attacker changes the timestamp, and sends the message $\{SR_j, M_{2j-1}, M_{2j}, T'_j, g_j, X_{jk}(1 \leq k \leq j)\}$ to $u_{j+1}^{(i)}$, if $|T'_{j+1} - T'_j| \leq \Delta T$ holds correct, $u_{j+1}^{(i)}$ computes SR'^{*}_j, B'^{*}_j, M'^{*}_{2j}.

$$\begin{aligned} SR'^{*}_j &= h(g_j \cdot SK_i \cdot PT'_j) \\ B'^{*}_j &= M_{2j-1} \oplus SR'^{*}_j \\ M'^{*}_{2j} &= h(B'^{*}_j || g_j || T'_j). \end{aligned}$$

From the above, $M'^{*}_{2j} \neq M_{2j}$ can be known, and the attacker can't know SK_i, the session is aborted. Therefore, the attacker cannot participate in the process of negotiating the key of the subgroup by replaying the information.

4 Conclusion

This paper proposes a dynamic multicast key management scheme based on a grouped linked list. In the scheme, the group is divided into several small subgroups, so that the burden generated by the key update is shared. The joining and exiting of group members will only affect the subgroups where they are located, which does not affect the entire multicast group, and is more suitable for the scalability requirements of large-scale dynamic multicast key management. Adopting the form of linked list, the number of communication rounds between users is reduced, which achieves high efficiency. Hash function and elliptic curve cryptography are used in key negotiation to improve the security of the whole scheme, which is more likely to succeed in satisfying the current security requirements of group communication. The proposed scheme can guarantee the confidentiality, forward security, backward security, and resist collusion attack and replay attack. The distributed key management mechanism is adopted between subgroups and members of subgroups, which reduces the storage of keys and makes it simpler and more efficient in the key updating process.

Acknowledgments

This work was supported by the National Key R&D Program of China on Blue Granary technology innovation (2019YFD0900805) and Shanghai Collaborative Innovation Center for Improved Breed Creation and Green Breeding of Aquatic Animals (A_1-3605-21-0002).

References

[1] M. Bilal, S. G. Kang, "A secure key agreement protocol for dynamic group," *Cluster Computing*, vol. 20, no. 3, pp. 2779-2792, 2017.

[2] M. Bilal and S. Pack, "Secure distribution of protected content in information-centric networking," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1921-1932, 2020.

[3] L. L. Cao, W. C. Ge, "A secure and efficient multifactor mutual certificateless authentication with key agreement protocol for mobile client-server environment on ECC without the third-party," *International Journal of Security and Its Applications*, vol. 10, no. 10, pp. 215-226, 2016.

[4] S. Q. Cao, W. R. Liu, L. L. Cao, *et al.*, "An improved anonymous authentication protocol for location-based service," *IEEE Access*, vol. 7, pp. 114203-114212, 2019.

[5] J. W. Chen, L. Xu, "New group rekeying scheme for multicast in ad hoc networks," *Computer Engineering*, vol. 24, pp. 164-167, 2007.

[6] Y. L. Chen, G. Yang, "EBS-based collusion resistant group key management using attribute-based encryption," *China Communications*, vol. 01, pp. 92-101, 2012.

[7] K. Y. Choi, J. Y. Hwang, D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," in *International Workshop on Public Key Cryptography*, pp. 130–144, 2004.

[8] L. Gao, C. M. Tang, Y. Q. Zhang, "Multi-parties key agreement protocol based on elliptic curve," *Chinese Journal of Network and Information Security*, vol. 2, no. 05, pp. 77-80, 2016.

[9] M. C. Gorantla, C. Boyd, Nieto, *et al.*, "One round group key exchange in the standard model," *IACR Cryptology ePrint Archive*, vol. 83, 2010.

[10] Y. S. Hu, H. Shan, T. Ma, "Group key management scheme design for heterogeneous wireless sensor network," *Computer Engineering*, vol. 37, no. 01, pp. 149–153, 2011.

[11] S. M. Hussein, Ramos, and Bermejo, "Distributed key management to secure IoT wireless sensor networks in smart-Agro," *Sensors*, vol. 20, no. 8, 2020.

[12] T. F. Lee, M. Chen, "Lightweight identity-based group key agreements using extended chaotic maps for wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10910-10916, 2019.

[13] B Majid, B. A. Mohammad, B Morteza, *et al.*, "Cryptanalysis and improvement of a user authentication scheme for internet of things using elliptic curve cryptography," *International Journal of Network Security*, vol. 21, no. 6, pp. 897-911, 2019.

[14] S. Mandal, S. Mohanty, B. Majhi, "CL-AGKA: Certificateless authenticated group key agreement protocol for mobile networks," *Wireless Networks*, vol. 26, no. 4, pp. 3011-3031, 2020.

[15] S. Mittra, "IOLUS: A framework for scalable secure multicasting," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 4, 1997.

[16] A. S. Pande, Y. Joshi, and M. Y. Joshi, "Analysis on logical key hierarchy and variants for secure group communication," in *Computing, Communication and*

Signal Processing, Advances in Intelligent Systems and Computing, pp. 419-430, 2019.

[17] S. Setia, S. Koussih, S. Jajodia, *et al.*, "Kronos: A scalable group re-keying approach for secure multicast," in *Proceeding IEEE Symposium on Security and Privacy*, pp. 215-228, 2000.

[18] M. Steiner, G. Tsudik, M. Waidner, "CLIQUES: A new apporach to group key agreement," in *Proceedings of 18th International Conference on Distributed Computing Systems*, IEEE, 1998.

[19] Y. Sun, S. Yin, J. Liu, and L. Teng, "A certificateless group authenticated key agreement protocol based on dynamic binary tree," *International Journal of Network Security*, vol. 21, no. 5, pp. 843-849, 2019.

[20] Z. G. Tan, H. P. Huang, R. C. Wang, *et al.*, "A key management scheme using ECC with grouping cluster," *Computer Technology and Development*, vol. 22, no. 02, pp. 176-180, 2012.

[21] P. Vijayakumar, R. Naresh, L. J. Deborah, *et al.*, "An efficient group key agreement protocol for secure P2P communication," *Security and Communication Networks*, vol. 9, no. 17, pp. 3952-3965, 2016.

[22] F. S. Wu, H. G. Zhang, "Key agreement protocol of non-signature authentication based on binary tree," *Journal of Computer Research and Development*, vol. 54, no. 12, pp. 2797-2804, 2017.

[23] H. Xie, L. M. Zuo,P. Z . Tang, "Multi-parties key exchange protocol based on elliptic curve cryptosystem," *Journal of Xinyang Normal University*, vol. 24, no. 04, pp. 533-535, 2011.

[24] J. Z. Xu, K. H. Liang, Y. X. Dong, "Study of distributed multicast key management based on creditability," *Application Research of Computers*, vol. 27, no. 01, pp. 271-273, 2010.

[25] Y. Yang, X. H. Zheng, and X. M. Liu, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system," *Future Generation Computer Systems*, vol. 84, pp. 160-176, 2018.

[26] H. Zhang, S. Y. Shi, Q. Shi, "Design of key management scheme for secure multicast based on LKH tree," *Computer Engineering and Design*, vol. 40, no. 02, pp. 312-316, 2019.

[27] Q. K. Zhang, Y. Gan, R. F. Wang, J. M. Zheng, and Tan, "Inter-cluster asymmetric group key agreement," *Journal of Computer Research and Development*, vol. 55, no. 12, pp. 2651-2663, 2018.

[28] H. Zhong, W. Y. Luo, J. Cui, "Multiple multicast group key management for the internet of people," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 3, 2017.

Biography

Cao Liling received the B.S. degree in electronic information science and technology from Central South University in 2004, and M.S. degree in physics electronics from Central South University in 2007, and the Ph.D. degree in testing technology and automation from Tongji University in 2017. Now she is a teacher of Department of Engineering Science and Technology Engineering in Shanghai Ocean University. Her main research is Network security, authentication protocol.

LIANG Mei received her bachelor's degree in electronics and information engineering from ChuZhou University in 2018. Now, she is a student at the school of engineering, Shanghai Ocean University. Her main research is communication security and Internet of things technology.

ZHANG Yu received her bachelor's degree in mechanical engineering from Huaiyin Institute of Technology in 2019. She received her MS degree in mechanical engineering in Shanghai Ocean University in 2021. Her main research is communication security and Internet of things technology.

CAO Shouqi received his bachelor's degree in mechanical manufacturing technology and equipment from Sichuan University in 1996. He received his MS degree in mechanical manufacturing and automation from Sichuan University in 1999. He received his postdoctoral degree in control science and Engineering in Shanghai University in 2009. Now he is a professor and doctoral supervisor of Department of Engineering Science and Technology Engineering in Shanghai Ocean University. His main research interest is marine Internet of things engineering, Fisheries Engineering and automation technology research.