A Hybrid Group Key Management Scheme for UAV – MBN Network Environment Increasing Efficiency of Key Distribution in Joining Operation

R. Mahaveerakannan[1(✉)] and C. Suresh Gnana Dhas[2]

[1] Information Technology, St. Peter's University, Chennai, India
mahaveerakannan10@gmail.com
[2] Vivekananda College of Engineering for Women, Namakkal, India
sureshc.me@gmail.com

Abstract. The advent of wireless Ad-Hoc networks has seen an associated emergence of group applications. In order to secure group applications and prevent unauthorized users from accessing communication data that cannot be protected by wireless Ad-Hoc networks and IP multicast alone, group communication content needs to be encrypted by a shared group key. Key management is necessary to ensure the safety of this group key and to protect the group communication. However, problems arise when group key management schemes are applied to the wireless environment relates to three issues: performance, security and network compatibility. In this paper, the focus is made on UAV-MBN, a military network with a proposal to group key management architecture for the UAV-MBN. In order to tackle two particular wireless group key management problems, operational efficiency and multiple-membership changes, a hybrid group key management approach - which operates within each theatre of UAV-MBN is included. By performing micro-key management, this approach can reduce the operational costs associated with key management and improve the operational efficiency of wireless group key management. A membership-oriented group key management scheme is concentrated to tackle the performance problem of multiple-membership changes. This approach, compared to traditional application-oriented group key management approaches, offers more effective management of multiple-membership changes.

Keywords: Centralized control · Decentralized control
Distributed management · Mobile ad hoc networks · Security management
Unmanned autonomous vehicles

1 Introduction

Group key management needs to identify a way to efficiently distribute keying materials to its group members and raise the operational efficiency, which is to be dealt with high priority. This becomes more critical when a group key management approach is applied in the wireless environment due to the resource limitations of both UAV-MBN networks and mobile devices. Several group key management approaches [7–10] exists

© Springer Nature Singapore Pte Ltd. 2018
G. P. Venkataramani et al. (Eds.): ICIIT 2017, CCIS 808, pp. 93–107, 2018.
https://doi.org/10.1007/978-981-10-7635-0_8

and are investigated. However, these encounter serious operational efficiency problems when they are applied in the wireless theatre based architectures similar to that of UAV-MBN. In this paper, a new wireless group key management approach named as hybrid group key management (HGKM) is planned, to be implemented in UAV-MBN network based on the proposed wireless group key management architecture. This new group key management approach, HGKM for UAV-MBN is designed to address the operational efficiency issues with the following assumptions:

- A small key management structure that enables micro-key management operations to achieve efficiency for both the key controller and group members;
- A combination of centralized and decentralized key management approaches that allows members to involve in key management in order to reduce the overhead for the key controller during rekeying;
- A simple message delivery scheme that supports reliable transmission of rekeying messages; and
- Key controllers that co-operate to facilitate rekeying during the handoff.

2 Existing Group Key Management Approaches

In the distributed group key management approaches such as Group Diffie-Hellman Key Exchange [1, 6] and Tree-based DH Key Management [8–10], there is no explicit key controller that manages the group key and supporting keys. The group key is generated in such a way that each member contributes its own parts to calculate the shared group key. A number of expensive exponential computations are required in the key generation, which cannot be afforded by mobile devices participating in the UAV-MBN. Furthermore, in order to calculate the group key, each member needs to broadcast its contribution within the group following a one-by-one transmission pattern, which is a lengthy pattern for large groups. Distributed group key management [11, 12] approaches consequently have a slow key converging speed during key generation. Two serious performance issues namely the expensive computation requirement and the slow speed of key generation prevents the distributed key management [13, 14] approaches from being applied to the UAV-MBN network. The centralized group key management schemes such as LKH [3–5] and OFT [2] also face operational efficiency problems when they are deployed in the UAV-MBN network. In centralized group key management, all group members are organized into a single hierarchical key tree. Due to operational efficiency issues, distributed and centralized group key management approaches are not suitable for application in the UAV-MBN network. A more efficient group key management approach is required for the UAV-MBN network.

3 Hybrid Group Key Management (HGKM)

Based on the analysis of various centralized group key management approaches, it has become apparent that the reason for operational inefficiency in centralized approaches is rooted in the organization of all group members in one single and large hierarchical

structure. In order to minimize the rekeying impact on the group members and to reduce the overhead of key management, micro-key management operations are proposed to be performed on small management structures as theatres called as operation units. An operation unit is a small fixed-sized key management structure. Each operation unit contains a hierarchical structure for the purpose of key management. Under the HGKM approach, all group members within a theatre form a key management group called theatre-key-management-group for the purpose of key distribution. The theatre-key-management-group is composed of operation units. There are two types of operation units in HGKM: *leader unit and member unit.*

The upper level, called the leader units level, is formed by leader unit(s). There are two roles in the leader unit: *leader and leadership candidate.* Leader refers to a group member in a leader unit who is designated as a leader of a member unit on the lower level. The responsibility of the leader is to assist the theatre key controller (TKC) to distribute the keying materials to the group members within its member unit. On the other hand, leadership candidate refers to a member in a leader unit who has not been assigned as a leader yet. Its responsibility is to work as a backup to the current leader. When a leader leaves the group, a candidate can be designated as the new leader of the affected member unit immediately. The purpose of having leadership candidates is to reduce the operational cost of rekeying when the leader leaves the group.

3.1 Generation of Leader and Member Units

In the initial stage of HGKM, a TKC first creates the leader unit. After the TKC completes the generation of leader unit(s), the TKC creates member units for the new incoming group members and designates a member from the leader unit to be the leader of the newly-generated member unit. During the generation of operation units, the TKC continually monitors the changes in the ratio of leadership candidates.

The Join Operation

Based on the proposed wireless group key management architecture when a user joins a group, the TKC needs to enforce backward secrecy to prevent the new joining member from decrypting previous group data by updating the theatre traffic encryption key (CTEK). Therefore, in HGKM, the join operation starts with the user sending the join request to the group key controller (GKC) for authentication and authorization. In addition, the incoming member also sends a request via Internet Group Management Protocol (IGMP) to the TKC requesting group communication data from the base station.

$$user \rightarrow GKC : \{group\ join\ request\}$$

$$user \rightarrow TKC : \{request\ for\ receiving\ the\ group\ communication\ data\}$$

After receiving the join request, the GKC validates the user. If authentication is successful, the GKC sends the new incoming member the group traffic encryption key (GTEK) encrypted with a pair-wise key known only to the GKC and the user, while the GKC informs the TKC that the user is authenticated and authorized.

$$GKC \rightarrow user : \{k_{GTEK}\}k_{GKC} - user$$
$$GKC \rightarrow TKC : \{user\ is\ authenticated\}$$

After receiving this control message from the GKC, the TKC contacts the new joining member and invokes the join procedure. There are three steps to perform the joint action in the theatre.

- Step1: the TKC assigns the new joining member into an operation unit, where the leader unit has priority over the member unit.
- Step 2: the TKC sends the new member a set of keys including the theatre traffic encryption key (CTEK) and the supporting keys it is entitled to know.
- Step 3: the TKC invokes the join rekeying procedure to update the keys for the remaining members of the theatre.

In HGKM, there are two types of join operations within the theatre: the joining leader unit and the joining member unit. The TKC assigns the new member into an operation unit according to the key management policy, where the leader unit is given priority over the member unit. The purpose of doing this is to ensure there are enough members working as leaders and leadership candidates in the leader units' level for key management because the responsibility of a leader is to assist the TKC to distribute rekeying messages within its member unit.

The Joining Leader Unit

Once the TKC finds an available empty slot in the leader unit, the TKC assigns the joining member into that slot. In order to ensure backward secrecy, the TKC needs to generate new keys to replace the affected current keys. After sending the newly-generated keys to the new member, the TKC invokes the rekeying procedure to update the affected keys for the remaining group members. This rekeying process follows the bottom-up procedure and is divided into two steps.

- Step 1: the TKC updates the keys for the directly-affected leader unit where the new member resides. This rekeying procedure is the same as that in LKH.
- Step 2: the TKC multicasts a rekeying message in the theatre-key-management group where all remaining members reside to update the CTEK for the group members within the theatre.

In order to better illustrate the joining leader unit, we provide an example, shown in Fig. 1, to further describe this rekeying procedure. In Fig. 1, when user 8 joins the group, the TKC assigns user 8 into leader unit 2 as a leadership candidate. The TKC generates the new keys $(k'_{CTEK}, k'_{78}, k'_{58})$ to replace the current keys $(k_{CTEK}, k_{78}, k_{58})$ to ensure backward secrecy. The TKC sends these new keys to user 8.

$$TKC \rightarrow user8 : \{k'_{CTEK}, k'_{78}, k'_{58}\}k_8$$

In this rekeying message, the total of $h_{unit} + 1$ keys is encrypted by the TKC, where h_{unit} is the height of the key tree for the operation unit. After rekeying the new joining user, the TKC invokes the rekeying procedure to update the keys for the remaining group members within the theatre. Following the rekeying procedure, in step 1, the

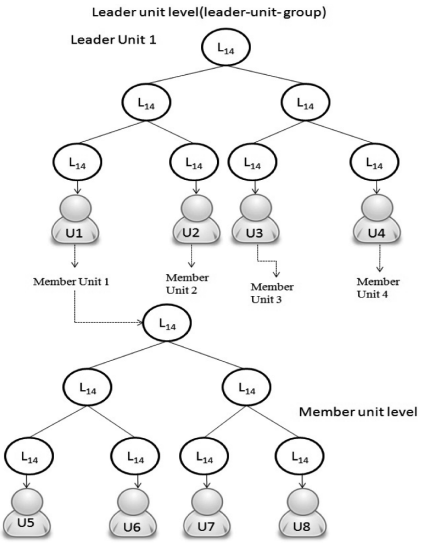


Fig. 1. Logical structure of HGKM

TKC updates the affected keys in leader unit 2 where the new member, user 8, has been assigned. The TKC generates two rekeying messages for user 7 and users 5 and 6 respectively to update the affected keys.

$$User\ 7 : \{k'_{78}, k'_{58}\}k_7$$
$$User\ 5, 6 : \{k'_{58}\}k_{56}$$

Due to the small size of these rekeying messages, the TKC places them into a single integrated rekeying message and multicasts it directly to the affected leader unit 2.

$$TKC \rightarrow \{leader\ unit\ 2\}$$
$$: \{for\{user\ 7\} : \{k'_{78}, k'_{58}\}k_7,\ for\ \{user\ 5, 6 : \{k'_{58}\}\ k_{58}\}$$

When the members in leader unit 2 receive this integrated rekeying message, each member can gain the latest keys from the corresponding section. In this step, the TKC sends a single rekeying message and the number of keys encrypted is:

$$1 + 2 + \ldots\ldots\ldots + h_{unit} = \frac{h_{unit}(h_{unit} + 1)}{2} \tag{1}$$

where h_{unit} is the height of the key tree for the operation unit.

After updating keys in leader unit 2, the TKC generates a rekeying message that contains the latest theatre TEK ($k'_{CTEK)}$) encrypted by the current CTEK ($k_{CTEK)}$) and multicasts it within the theatre-key-management-group that accommodates all the group members within the theatre.

$$TKC \Rightarrow \{theatre - key - management - group\} : \{k'_{CTEK}\}k_{CTEK}$$

When the remaining group members receive this message, they can gain the new CTEK for the future group communication. During the rekeying procedure of the joining leader unit, it can be observed that the TKC sends 3 rekeying messages. One is for the new joining member, one is for the directly-affected leader unit where the new member resides and the last is for all remaining members. In these three rekeying messages, the total number of keys encrypted by the TKC is

$$(h_{unit} + 1) + (1 + 2 + \ldots\ldots\ldots + h_{unit}) + 1 = \frac{(h_{unit} + 1)(h_{unit} + 2)}{2} + 1 \qquad (2)$$

where h_{unit} is the height of the key tree for the operation unit.

During the rekeying, the members in the directly-affected leader unit receive 2 rekeying messages; the first is the integrated message containing all the keying materials for leader unit 2 while the other contains the new CTEK for all remaining group members. The members outside the directly-affected leader unit only receive one single rekeying message to update the CTEK.

The Joining Member Unit

If the TKC cannot find an available slot in the leader unit(s) for the new joining member, then the incoming user is assigned to a member unit. The joining procedure is similar to the joining leader unit. The TKC generates new keys to replace the affected current keys in the directly-affected member unit and sends these new keys to the new joining user. After this, the TKC invokes the rekeying procedure to update keys for the remaining group members. Two steps similar to those performed for the joining leader unit are involved.

- Step 1: the TKC updates the keys for the directly-affected member unit where the new joining member has been assigned. The TKC generates and multicasts an integrated rekeying message which contains all the rekeying messages required by the directly-affected members unit.
- Step 2: the TKC multicasts a rekeying message that contains the new CTEK encrypted by the current CTEK to the theatre-key-management group to update CTEK for all remaining group members.

Considering a scenario Fig. 2, when user 9 joins the group, the TKC assigns it into member unit 2. The TKC generates the new keys (k'_{CTEK}, k'_{910}, k'_{912}) and sends them to new joining member, user 9.

$$TKC \rightarrow user9 : \{k'_{CTEK},\ k'_{910},\ k'_{912}\}k_9$$

In this rekeying message, it can be observed that the number of keys encrypted by the TKC is $h_{unit} + 1$, where h_{unit} is the height of the key tree for the operation unit. After rekeying user 9, the TKC invokes the rekeying procedure to update the keys for the remaining group members. First, in step 1, the TKC updates the keys for the directly-affected operation unit, member unit 2. The TKC refreshes the current keys for users 10, 11, 12 and user 2, the leader of a member unit 2 by multicasting an integrated rekeying message.

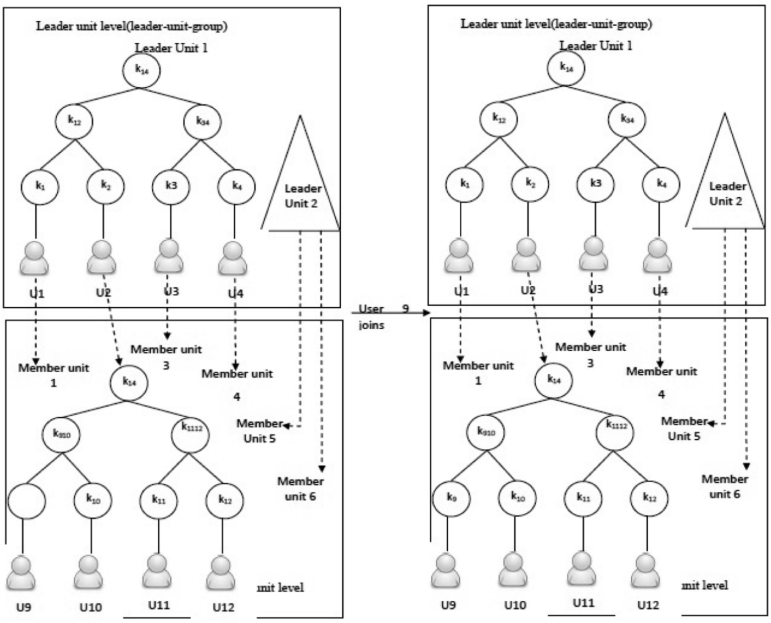


Fig. 2. New user 9 joins the group

$$TKC \rightarrow \{member\ unit\ 2\} : \{for\{user\ 10\} : \{k910', k912'\}\ k10$$
$$for\ \{user\ 11, 12 : \{k'_{912}\}\ k_{1112}\}$$
$$for\ \{user\ 2 : \{k'_{912}\}\ k_2\}$$

After receiving this message, users 2, 10, 11, 12 can update their keys from the corresponding section. For the purpose of reliable delivery, the leader of the member unit 2, user 2, stores this rekeying message for a period of time until it receives new keying materials from the TKC. In this process, the TKC sends just one rekeying message and the number of keys encrypted by the TKC during this step is:

$$(1 + 1 + 2 + \ldots\ldots\ldots + h_{unit}) = \frac{h_{unit}(h_{unit} + 1)}{2} + 1 \tag{3}$$

where h_{unit} is the height of the key tree for the operation unit.

After rekeying the directly-affected member unit, in step 2, the TKC multicasts a rekeying message, which contains the new CTEK, k'_{CTEK}, to the theatre-key-management-group to update the CTEK for all remaining group members within the theatre.

$$TKC \Rightarrow \{theatre - key - management - group\} : \{k'_{CTEK}\}k_{CTEK}$$

Three rekeying messages are sent by the TKC during the whole rekeying process. One message is for the new joining group member, one is for the directly-affected

member unit and the final message is for the remaining members. The number of keys encrypted by the TKC is:

$$(h_{unit}+1)(1+1+2+\ldots\ldots\ldots+h_{unit})+1=\frac{(h_{unit}+1)(h_{unit}+2)}{2}+2 \qquad (4)$$

where h_{unit} is the height of the key tree for the operation unit.

During this rekeying, the leader and the members of the directly-affected member unit receive two rekeying messages: the integrated rekeying message and a message updating the CTEK for all remaining group members. The group members outside the directly-affected member unit only receive one rekeying message updating the CTEK.

From Table 1, The number of messages sent by the TKC and the number of keys encrypted by the TKC during the rekeying process in two different types of join operations is given as; h_{unit}: the height of the key tree for operation unit in HGKM

Table 1. Joining a leader and member unit

	Number of messages sent out by TKC	Number of keys encrypted by TKC
Joining leader unit	3	$\frac{(h_{unit}+1)(h_{unit}+2)}{2}+1$
Joining member unit	3	$\frac{(h_{unit}+1)(h_{unit}+2)}{2}+2$

4 Performance Analysis

Operational efficiency is the highest priority in any wireless group key management scheme due to the resource limitations of both UAV-MBN networks and mobile devices. A wireless group key management approach cannot be recognized as efficient and practical if it cannot meet the requirements of operational efficiency. Therefore, in this section, we analyse and evaluate the operational costs of key management in HGKM to demonstrate that HGKM is an efficient and practical wireless group key management approach suitable for deployment in the UAV-MBN network. In order to evaluate the performance of HGKM, we set the centralized group key management approaches LKH and OFT as the benchmarks. The reasons for selecting these as benchmarks are: (i) LKH and OFT can be applied within the theatre; (ii) LKH and OFT also apply a hierarchical key structure to manage keying materials which are similar to HGKM; and (iii) LKH and OFT are considered to be two of the most widely-used and efficient group key management approaches because the operational costs of these are $O(log_d n)$ where d is the degree of the key tree and n is the total number of group members.

4.1 Communication Cost

The theatre key controller (TKC) is the main key management entity in HGKM. Consequently, the communication cost generally refers to the communication overhead of the TKC during the rekeying procedure caused by the join and leave procedures. The communication cost can be measured by the number of rekeying messages transmitted by the TKC during rekeying. Without loss of generality, we apply a binary tree to build the key management structure in HGKM, LKH, and OFT, as a binary tree is easy to create, manage and maintain.

The Communication Cost of the Join Operation

In HGKM, there are two kinds of join actions: joining the member unit and joining the leader unit. From Table 2, it can be observed that the communication cost of the join action in LKH and OFT is proportional to the size of the whole group. Along with the increased group size, the communication cost of the join procedure in LKH and OFT also becomes larger. In contrast, the communication cost of the join procedure for the TKC in HGKM is a constant value that is achieved by applying an integrated rekeying message to contain all the rekeying information for the affected operation unit. In HGKM, micro-key management is performed within a small fixed-sized operation unit. The number and the size of rekeying messages sent to the affected operation unit are therefore minimal. Because HGKM has the ability to place the rekeying messages into one single integrated message, it can utilize the capacity of multicast transmission to improve the throughput and efficiency of network transmission. We apply the theory of expectation value to calculate the average communication cost of a join action in HGKM.

Table 2. The communication cost of the join action for TKC

Group key management algorithm		Communication cost
HGKM	Join member unit	3
	Join Leader unit	3
LKH		$h + 1$
OFT		$h + 1$

$$Cost_{communication}(join) = Cost_{joining_leader_unit} \times P_{joining_leader_unit} + Cost_{joining_member_unit} \times P_{joining_member_unit} \quad (5)$$

where $Cost_{joining_leader_unit}$ and $Cost_{joining_member_unit}$ are the communication costs of joining leader unit and joining member unit respectively.

$p_{joining_leader_unit}$ and $p_{joining_member_unit}$ present the probability of joining leader unit and joining member unit correspondingly.

In HGKM, the probability of joining leader unit and joining member unit join are:

$$p_{joining_leader_unit} = \frac{n_{members_in_leader_units}}{n_{total_group_members}}$$
$$p_{joining_member_unit} = \frac{n_{members_in_member_units}}{n_{total_group_members}}$$

where $n_{members_in_leader_units}$ is the number of members in the leader units, $n_{members_in_member_units}$ is the number of members in the member units and $n_{total_group_members}$ is the total number of members in the whole group.

- The communication cost of the join action in HGKM is a constant value independent of the size of the group;
- The communication cost of the join action in LKH and OFT is three to five times higher than that of HGKM. Moreover, the communication cost of the join action in LKH and OFT increases with the growth of the number of group members;
- The gap in the communication cost between HGKM and LKH and OFT becomes increasingly wider as the group size increases.

These three features ensure that HGKM can achieve much better communication efficiency in key management during the join procedure than the LKH and OFT approaches.

4.2 Computation Cost

Computation cost is another important parameter that can be used to evaluate the operational efficiency of group key management. The task of encryption and decryption is the heaviest work done by the TKC and group members during rekeying, making it an appropriate criterion for assessment. For the TKC, this parameter can be measured by the number of keys encrypted during the rekeying procedure. For group members, this parameter can be approximately assessed by ascertaining the number of rekeying messages received by members. During rekeying, every rekeying message is multicasted to all group members to improve communication efficiency. Each member needs to process every single received message to determine if it is the intended recipient. The computation cost for the member is therefore directly related to the number of received messages. In the next two sections, we analyse the computation costs of the join and leave operation respectively. We set the costs of LKH and OFT as the benchmarks for the purpose of comparison.

The Computation Cost of the Join Operation

(i) The computation cost of the join operation for the TKC Table 3 summarizes the computation cost of a single join operation in HGKM, LKH, and OFT, where the formulas are from the Sect. 3.

Table 3. The computation cost of the join operation for TKC

Group key management	Computation cost of join action for TKC	
	Leader unit	Member unit
HGKM	$\frac{(h_{unit}+1)(h_{unit}+2)}{2}+1$	$\frac{(h_{unit}+1)(h_{unit}+2)}{2}+2$
LKH	$\frac{(h+1)(h+2)}{2}-1$	
OFT	$2\ h$	

h_{unit}: the height of the key tree for operation unit in HGKM
h: the height of the group key tree in LKH and OFT

In Table 3, it can be observed that the computation cost of the join action in HGKM is proportional to the power of the height of the key tree for the operation unit, $2O\ (h_{unit})$. This cost is independent of the size of the group members. Once the size of the operation unit is determined, the computation cost of the join operation for HGKM becomes a constant value. In contrast, in terms of LKH, the computation cost of the join operation is proportional to the power of the height of the group key tree, $O(h2)$, while the computation cost of the join action for OFT is in ratio to the height of the group key tree, $O(h)$, which is due to the dependent key generation bypassing the old key through a one-way function to obtain a new key.

In Sect. 3, we introduced two join scenarios in HGKM: joining leader unit and joining member unit. Based on the theory of expectation value, for a single join action, the average computation cost is:

Where p_{leader_unit} (join) and p_{member_unit} (join) are the probability of joining leader unit and the probability of joining member unit respectively. In HGKM, the probability of joining leader unit and joining member unit are:

$$p_{leader_unit}(join) = \frac{n_{member_in_leader_units}}{n_{total_group_members}}$$
$$p_{member_unit}(join) = 1 - p_{leader_unit}(join)$$

where $n_{members_in_leader_units}$ is the number of members in the leader units and $n_{total_group_members}$ is the total number of member in the group. Therefore, the average computation cost of the join action for HGKM can be simplified as follows:

$$\begin{aligned} Cost_{computation}(join) &= \left(\frac{(h_{unit}+1)(h_{unit}+2)}{2}+1\right) \times p_{leader_{unit}}(join) \\ &\quad + \left(\frac{(h_{unit}+1)(h_{unit}+2)}{2}+2\right) \times p_{member_unit}(join) \\ &= \frac{(h_{unit}+1)(h_{unit}+2)}{2} + 2 - \frac{n_{member_in_leader_units}}{n_{total_group_members}} \end{aligned}$$

The comparison of computation cost of join for the TKC in HGKM, LKH, and OFT is shown in Fig. 3. In this example, we assume that the size of the operation unit is 32.

In Fig. 3, three features of the computation cost of the join action for the TKC can be observed.

- Due to the application of dependent key generation, the computation cost of the join action for the TKC in OFT is low and proportional to the height of the group key tree, $O(h)$.
- HKGM can achieve the same level efficiency as that of OFT because there keying process is confined within the small-sized operation unit. The computation cost of the join action for HGKM is insensitive to increasing group size. Despite the growth in the number of group members, the computation cost of the join operation for HGKM is little changed. The reason for this is the application of micro-key management within the scope of the operation unit;

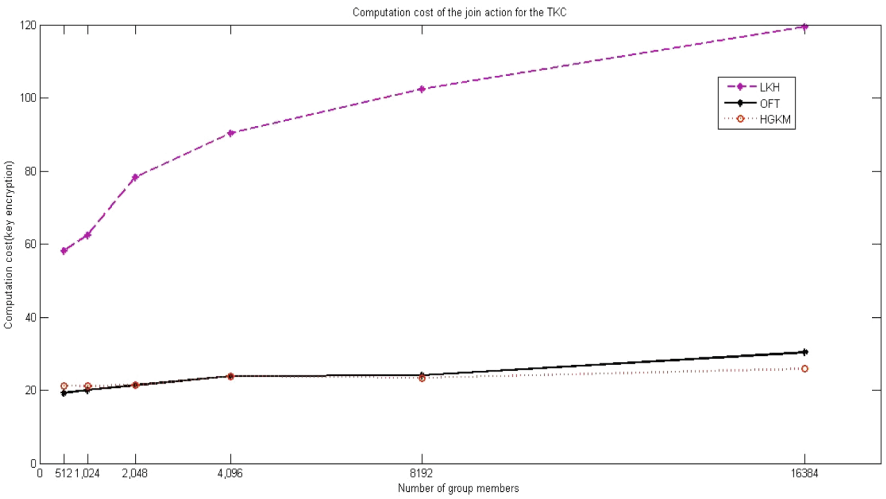


Fig. 3. The computation cost of the join action for the TKC

- LKH has the highest computation cost of the three approaches, with a computation cost of the join action two to six times higher than that of HGKM.

Moreover, the computation cost of the join action for LKH increases logarithmically as the group grows. The gap in the computation cost between LKH and HGKM consequently becomes increasingly wider. In conclusion, HGKM has an advantage over LKH in the computation cost of the join action, with the cost being similar to that of OFT. Moreover, the advantage increases with the growth of the group size.

(ii) The computation cost of the join operation for members For group members, the computation cost can be measured by the number of received rekeying messages. As discussed in Sect. 3, in the centralized group key management approaches, each rekeying message is multi-casted to the whole group. Furthermore, due to the small number and size of these rekeying messages, they can be placed into an integrated message for more efficient transmission. As a result of this transmission, members in the directly-affected operation unit only need to process this integrated message to find the relevant information. For the purpose of comparison, in HGKM, we assume that the computation cost of the join action for the members in the directly-affected operation unit is the number of rekeying messages contained in the integrated package. Table 4 summarizes the computation cost of the join action for members in HGKM, LKH, and OFT, where the formula is from the previous Sect. 3.

Figure 4, illustrates the computation cost of join on the members' side for HGKM, LKH, and OFT. In this example, we suppose that the size of the operation unit is 32.

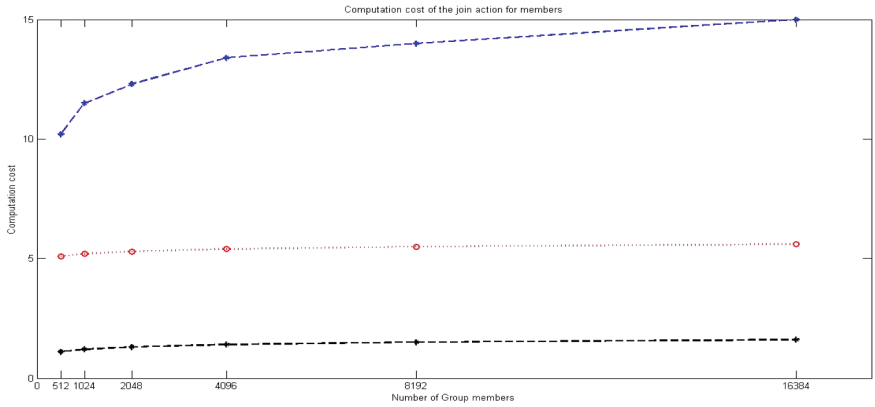


Fig. 4. The computation cost of the join action for members

Table 4. The computation cost of the join operation for group members

Group key management algorithm	Joining leader unit		Joining member unit	
	Members of the directly affected operation unit	Members outside the directly - affected operation unit	Members of the directly affected operation unit	Members outside the directly-affected operation unit
HGKM	$h_{unit} + 1$	1	$h_{unit} + 1$	1
LKH	H			
OFT	H			

h_{unit}: the height of the key tree for operation unit in HGKM
h: the height of the group key tree, which equals 2 log n, n is the total number of group members.

From Table 4, it can be observed that the computation cost of join for the members in the directly-affected operation unit is proportional to the height of the key tree for the operation unit in both join scenarios. In contrast, the computation cost of the join action for LKH, and OFT equals the height of the group key tree. Compared to the size of the key the tree for LKH, and OFT, HGKM has small operation units. HGKM consequently has better computation efficiency than LKH and OFT. In addition, the most important improvement in the computation cost for HGKM is that the computation cost of the join action for group members outside the directly-affected operation unit is just one. This achievement minimizes the impact of rekeying on the remaining members and drastically improves the computation efficiency for members.

From Table 5, it is observed that the computation cost of the join action for the members in the directly-affected operation unit in HGKM is a constant value and independent of the group size. The reason for this is that the height of the key tree for the operation unit is determined by the size of an operation unit.

Table 5. The operation cost for HGKM

HGKM		Operation Cost
Communication	Join(TKC)	3
Computation	Join (TKC)	$\frac{(h_{unit}+1)(h_{unit}+2)}{2} + 2 - \frac{n_{member_in_leader_units}}{n_{total_group_members}}$
	Join (member)	Members in the directly –affected operation unit: $h_{unit}+1$
		Members outside the directly –affected operation unit:1

h_{unit}: height of key tree for the operation unit in HGKM
$n_{member_in_leader_units}$: total number of members in leader units
$n_{total_group_members}$: total number of members in HGKM

5 Conclusion

In order to tackle the problems of operational efficiency in wireless group key management, we have proposed a novel group key management approach: hybrid group key management (HGKM) in this chapter. This approach is specifically designed for the UAV-MBN network. The major contribution of HGKM is that HGKM performs micro-key management within a small area known as an operation unit. In HGKM, group members are organized into a number of operation units for the purpose of key management. Based on these operation units, micro-key management is performed. The features that allow HGKM to achieve operational efficiency can be summarized as follows: (1) key management can be restricted within the small scope of the operation unit to reduce operational costs from the perspectives of communication, computation and key storage; (2) HGKM combines the features of centralized and distributed key management approaches to allow members to involve in the key management in order to reduce the operational cost of the TKC during the rekeying process; and (3) HGKM has a simple and built-in reliable message delivery scheme based on the operation unit to provide reliable transmission of keying materials. Based on the analysis, evaluation, and comparison presented in this chapter, we conclude that HGKM can improve operational performance from the perspectives of communication, computation and key storage. HGKM offers an efficient and practical wireless group key management approach for the UAV-MBN network.

References

1. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. In: Proceedings of 3rd ACM Conference on Computer and Communications Security, pp. 31–37, March 1996
2. McGrew, D.A., Sherman, A.T.: Key establishment in large dynamic groups using one-way function trees: TIS. IEEE Trans. Softw. Eng. **29**(5), 444–458 (2003)
3. Wallner, D., Harder, E., Agee, R.: Key management for multicast: issues and architectures. RFC 2627, June 1999
4. Wong, C.K., Gouda, M., Lam, S.S.: Secure group communications using key graphs. In: Proceedings of ACM SIGCOMM, pp. 68–79, October 1998

5. Wong, C.K., Gouda, M., Lam, S.S.: Secure group communications using key graphs. IEEE/ACM Trans. Netw. **8**(1), 16–30 (2000)
6. Steiner, M., Tsudik, G., Waidner, M.: Key agreement in dynamic peer groups. IEEE Trans. Parallel Distrib. Syst. **11**(8), 769–780 (2000)
7. Rafaeli, S., Hutchison, D.: A survey of key management for secure group communication. ACM Comput. Surv. **35**(3), 309–329 (2003)
8. Kim, Y., Perrig, A., Tsudik, G.: Communication-efficient group key agreement. In: Proceedings of 17th International Information Security Conference, pp. 229–244, July 2004
9. Kim, Y., Perrig, A., Tsudik, G.: Group key agreement efficient in communication. IEEE Trans. Comput. **53**(7), 905–921 (2004)
10. Kim, Y., Perrig, A., Tsudik, G.: Tree-based group key agreement. ACM Trans. Inf. Syst. Secur. **7**(1), 60–96 (2004)
11. Sun, Y., Trappe, W., Ray Liu, K.J.: A scalable multicast key management scheme for heterogeneous wireless networks. IEEE/ACM Trans. Netw. **12**(4), 53–666 (2004)
12. Kim, H., Chung, B., Lee, Y., Park, Y., Yoon, H.: Weakness of the synchro-difference LKH scheme for secure multicast. IEEE Commun. Lett. **11**(9), 765–767 (2007)
13. Park, M.-H., Park, Y.-H., Jeong, H.-Y., Seo, S.-W.: Key management for multiple multicast groups in wireless networks. IEEE Trans. Mob. Comput. **12**(9), 1712–1723 (2013)
14. Mapoka, T.T., Shepherd, S.J., Abd-Alhameed, R.A.: A new multiple service key management scheme for secure wireless mobile multicast. IEEE Trans. Mob. Comput. **14** (8), 1545–1559 (2015)