EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

The energy-efficient group key management protocol for strategic mobile scenario of MANETs

Xiao Wang*, Jing Yang, Zetao Li and Handong Li

Abstract

Network security and its energy efficiency are facing tougher challenges for mobile *ad hoc* networks (MANETs) due to emerging purposive strategic internal attacks conducted by smart malicious nodes and unavoidable external attacks. Most of the current works investigated the group key management for scalability and efficiency performance in MANETs to defend the external attacks, while some works envisioned the intrusion detection system or trust management to defend the internal attacks. However, fewer related protocols or algorithms could combine them well to enhance dual security performance as well as energy efficiency in MANETs. To this end, we have proposed a novel group key management protocol with high energy efficiency for the strategic mobile scenario of MANETs, which is provided with three functions to address the issues of improving security and energy efficiency performance: (1) designing a self-organized group establishing algorithm for strategy mobile application scenarios to ensure stable groups in spite of users' mobility with reducing the cost of rekeying operation, (2) proposing a lightweight contributory key agreement and authentication mechanism based on the group Diffie-Hellman protocol for enhancing global security, and (3) researching a strategic mobile management mechanism based on the Prufer codec method handling the effect of mobility impacts to enhance the multicast energy efficiency and provide secret communication among roaming users in MANETs. Both theoretical analyses and simulation results have demonstrated that our protocol is more energy-efficient for strategy mobile application scenario of MANETs with a large number of users.

Keywords: MANETs; Group key management; Strategic mobile scenario; Mobile management model

1 Introduction

Mobile *ad hoc* networks (MANETs) are composed of a set of mobile nodes wirelessly connected without a support of any fixed infrastructure. Since MANETs can be rapidly accessed and flexibly deployed, it has revealed a growth potential in applications such as battlefield monitoring, disaster recovering, emergency handling, vehicular networking, etc. These applications for MANETs have been restricted by their own inherent natures, including constrained available resources, exposed communication medium, intermittent end-to-end links, and frequent changes in topology due to users' mobility, which are all prone to incur various kinds of attacks [1,2]. As a consequence, energy-efficient and secure group communication must be a prior concern for MANETs to ease these rigorous security threats.

To resist internal attacks and preserve energy in MANETs where mobile users cooperate with each other in a group to fulfill an assigned task, the object is to quickly detect the malicious nodes and evict them from the network. A lot of researches were done in the past but the most significant contributions were intrusion detection system (IDS) techniques [3-5] and trust-based managements [6-12], but fewer of the protocols made a decent trade-off between security and performance if users were in the strategic mobile scenario of

* Correspondence: ee.xiaowang@gzu.edu.cn
The Electrical Engineering College, Guizhou University, Guiyang 550025, People's Republic of China

MANETs. What is more, in order to avoid being caught and grab network resources for maximizing attack effect, some malicious nodes would elaborately choose a frequency at which they participate in the cooperation to cheat normal nodes. Consequently, current malicious nodes have changed conventional pure attack modes into purposive strategic attack modes [13], such as selective forwarding attack, selfish packet dropping attack, etc. All these purposive strategic attacks eventually run out of the throughput which results in network crashes if malicious nodes are not correctly detected out from the network.

To resist external attacks in distributed group communication networks, an effective way is to use cryptosystems. Among all these issues in MANETs, the group key management [14-18] is the most critical one. An algorithm which copes with the establishment, distribution, and maintenance of group keys is defined as a group key management protocol. Group key management aiming to enhance security is a very well-studied investigation in traditional wired networks and more recently in MANETs. However, most existing group key managements designed for traditional wired and wireless networks are not suitable for MANETs, especially for MANETs used in strategic mobile application scenarios, because, on the one hand, most algorithms are lacking a self-organized grouping mechanism for ensuring the scalability and performance of key management and, on the other hand, the effect of mobility causing dynamic topology changes on energy performance of the algorithms has not been considered. In this paper, aiming at a strategy mobile application scenario of MANETs, we first proposed a self-organized group establishing algorithm which only requires users interacting with their neighbors to let MANETs divide into several groups for satisfying the scalability. Second, we adopted the contributory key management and designed a highly efficient mobile management model to deal with the mobile issues when executing group operations. Third, we provided an efficient multicast mechanism using the Prufer codec algorithm to achieve the confidential communication among roaming users, which plays a significantly important role on cutting down the cost of group rekeying, improving scalability and efficiency of key management.

The contributions of our paper are as follows: First, we propose an online self-organized group establishing algorithm for strategic mobile scenarios of MANETs to realize the maneuverability putting the secure group operation into effect. Second, we design and propose a lightweight contributory key agreement and authentication mechanism based on the group Diffie-Hellman key management protocol in MANETs. Finally, we build a strategic mobile management model based on the Prufer codec algorithm handling the energy effect of mobility issues to enhance the multicast efficiency and provide secret communication among roaming users in MANETs.

2 Related works

A possible way to resist malicious nodes is to use behavior-based detection technology performed by normal network nodes through overhearing the communication in their neighborhood. This leverages the open broadcast nature of wireless communication. A popular instantiation of this technology is trust management approach in MANETs. The issue of trust in MANETs has been looked at by many researchers [6-11]. They usually use mathematical methods like the Dempster-Shafer belief theory for incorporating secondhand information (reports by other nodes) to create a reputation score of a node. Many reputation/trust-based approaches [10] suffer from poor protection against ballot stuffing or bad mouthing, which can be conducted as complex purposed strategic attacks by the smart malicious nodes we referred previously. The trust-based approaches are susceptible to behavior where a node functions correctly but provides wrong information about another node. Moreover, the approaches can suffer from non-convergent behavior, whereby the reputation of a good node gets stuck at a low value or that of a malicious node is falsely elevated. Moreover, in realistic MANETs, since the topologies change dynamically because of node movement, the trust management approach as well as the wider class of behavior-based detection cannot detect combination attacks. Additionally, they often mistakenly detect and isolate a legitimate member. Thus, how to properly model attack modes under imperfect monitoring and identify purposive strategic malicious attacks are still expected to be solved.

Some recent works [19-22] have studied the detection of such purposive strategic attack modes. Khalil and Bagchi [20] modeled similar kinds of purposive strategic attack models as stealth attacks which contain a suite of four attacks, including misrouting, power control, identity delegation, and colluding collision for wireless *ad hoc* networks. Then they proposed a protocol, SADEC, which could detect and isolate stealthy packet dropping attack efficiently. SADEC put forward two techniques that can be overlaid on baseline local monitoring: making the neighbors maintain additional information about the routing path and adding certain checking responsibility to each neighbor. In [21], an algorithm based on the non-parametric Kruskal-Wallis test was investigated to detect malicious nodes without any priori knowledge. The algorithm

made use of the current statistical differences between the cooperative decision and the non-cooperative decisions. However, these works could not make a decent trade-off between security and energy consumption and were not suitable for the strategic mobile scenario of MANETs.

In MANETs, the energy performance is significantly degraded as the scale of the network grows. For enhancing scalability, little work researches both security and self-organized performance issues of group establishing algorithm for MANETs. In [23], a grouping algorithm called MobHid, which guarantees longer lifetime of the group structure, was proposed. The mechanism was that it accurately predicts the mobility of each mobile host based on the stability of its neighborhood. This information is then used for establishing each group from hosts that will remain neighbors for a sufficiently long time, ensuring the formation of groups that are highly resistant to host mobility. In [24], two grouping algorithms were proposed to find the weakly connected dominating set (WCDS) for grouping the wireless *ad hoc* networks. One is a centralized approximation algorithm called DLA-CC based on distributed learning automata (DLA) for searching a near-optimal solution to the minimum WCDS problem. The other is a DLA-based algorithm called DLA-DC for grouping the wireless *ad hoc* networks which is a distributed implementation of DLA-CC. Using DLA-DC, the dominator nodes and their closed neighbors assume the role of the group managers and group members, respectively. In [25], a localized learning automata-based clustering algorithm called LLACA for wireless *ad hoc* networks was proposed. The proposed clustering method is a fully distributed algorithm in which each node chooses its group manager based solely on local information received from neighbors. For enhancing the security of the grouping algorithm, a trust-oriented grouping scheme was proposed [14]. The authors showed that trust is a relevant grouping criterion which could be extended to use for enforcing authentication and could be easily disseminated by the mobility of nodes in MANETs.

Several recent works investigated the group key management for scalability and energy efficiency performance in MANETs as in [15-18,26,27]. In [15], Zhu et al. studied the flexibility and scalability of dealing with risks in the practical usage of *ad hoc* networks and proposed a hierarchical scheme based on threshold cryptography to address both security and efficiency of key management in MANETs. In [16], Cho et al. proposed a scalable and efficient group key management protocol for secure group communications in MANETs and identified the optimal settings of the key management protocol to minimize the network traffic as well as to efficiently balance inter-group vs. intra-regional group key management overheads. Based on it, in [17], Cho et al. integrated the above group key management protocol with intrusion detection to handle both outsider and insider security attacks for group communication systems (GCSs) in MANETs. Aiming to improve both scalability and survivability of group key management for large-scale MANETs, in [18], Huang and Medhi presented a secure group key management scheme for hierarchical MANETs, which contains a multilevel security architecture based on the Bell-LaPadula model and a decentralized group key management infrastructure. However, their works assumed a fixed group size or allocated an optimal group size without considering an online self-organized grouping mechanism for efficient performance. Further, the effect of mobility causing dynamic topology changes on energy performance of the algorithms has not been considered well. It is not suitable for MANETs where users are in strategy mobile application scenarios.

To the best of our knowledge, fewer existing works consider the combination of group key management with high scalability, energy efficiency, and survivable group communication system in MANETs. In addition, to build a strategic mobile management model based on the Prufer codec algorithm handling the effect of mobility issues, our work is the first that considers the strategy mobile application scenarios for MANETs and designs an online self-organized group establishing algorithm of such scenarios to enhance the maneuverability putting the secure group operation in MANETs into effect.

3 Preliminary

3.1 Group establishing algorithm for strategic mobile scenario in MANETs

3.1.1 Strategic mobile scenario in MANETs

In the real MANET application scenes, the mobile properties of nodes can be classified into two types. The first one is called random mobile, such as vehicle-mounted mobile communication, wildlife monitoring, etc. Since the mobile paradigm and direction are unpredictable which may lead to a frequent rekeying process, it will cost too much energy if a contributory key agreement protocol is adopted. The second one is strategy mobile, such as battlefield monitoring, community mobile terminal tracking and networking, etc. The mobile of nodes is aimed at pursuing more communication links and more robust quality of communication services, that is to say, the probability of keeping moving on for a node which owns more stable links is lower than that for one with less stable links.

Consequently, the number of neighbor nodes and the quality of communication links have so much influence on the mobile attribute of nodes. The more the neighbor nodes are, the more reliable the communication quality is; hence, the probability of tending to be stable is greater and vice versa. In this paper, we mainly investigate and propose a group establishing algorithm among users in such strategic mobile scenario of MANETs.

3.1.2 The design principle of group establishing algorithm in strategic mobile scenario

Firstly, it is necessary to propose an algorithm to adaptively detect the number of neighbor nodes and communication quality around each user. On the basis of our previous research, we have put forward an approach of quality prediction and detection for adaptive links based on time slots [28]. With the help of this approach, the MANET topology and link quality can be accurately calculated within one time slot.

Secondly, the group establishing algorithm must follow the principle of self-organization for strategic mobile scenario. To this end, our algorithm will locally elect the group manager (i.e. GM for short) who is responsible for the establishment, distribution, and management of group keys; thus, the mobile state of the GM has to be relatively stable. Through establishing groups for users in the strategic mobile scenario of MANETs, there are more high-quality links between the members of the local group than between the users of other groups.

Thirdly, group establishing is the premise and foundation of group key management. In consideration of the resource constraints of MANET users, the group establishing algorithm should attempt to avoid frequent interaction among users and rapidly organize the mass mobile users into several communication groups in order to reduce the energy consumption.

Finally, the maintenance and updating of groups should satisfy the scalability of MANETs, i.e., the increase of users cannot exhaust storage and computation resources which results in performance crashes of related protocols during runtime.

3.1.3 Group establishing algorithm

3.1.3.1 Group rule Table 1 lists the notations used in this section.

We model the MANET topology as $T(U, L)$, where U is the set of all nodes and L is the set of communication links. For arbitrary $u_i \in U (i = 1, 2, \cdots, n)$, we define the open set of neighbor nodes as $N_u^{\text{open}} = \{v | \{u, v\} \in L\}$ and correspondingly we define the close set as $N_u^{\text{close}} = N_u^{\text{open}} \cup \{u\}$. The density of node u is obtained by modulus operation as $\text{Den}(u) = \left|N_u^{\text{open}}\right|$, i.e., the number of neighbors of node u. In our definition, all eligible groups constitute the set represented as $G = \{G_1, G_2, \cdots, G_s\}$, and it must simultaneously satisfy the following three conditions:

1. $\{G_1\} \cup \{G_2\} \cup \cdots \cup \{G_s\} = U$;
2. $\forall i,j = 1, 2, \cdots, s$ and $i \neq j$, $\{G_i\} \cap \{G_j\} = \Phi$;
3. $\forall u \in G_i (i = 1, 2, \cdots, s)$, $\left|N_u^{\text{close}} \cap G_i\right| \geq \left|N_u^{\text{open}} \cap (U - G_i)\right|$.

The above group conditions demonstrate that the number of communication links among all the members in any group is not less than that between the members in this group and other members out of this group.

3.1.3.2 Functions To describe our group establishing algorithm, we first introduce the following required functions:

Function 1: the minimum density set function $D_{\min}(S)$, where S is the input as well as the subset of set U, i.e., $S \in U$; the output of $D_{\min}(S)$ is one certain element of S satisfying the following equation:

Table 1 Notations of group establishing algorithm

Notations	Meanings
$N_i^{\text{open}}/N_i^{\text{close}}/\text{Den}(i)$	Open set of neighbors of node i/closed set of neighbors of node i/density of node i
a_i	The number of neighbors of node i which are in the same group with node i
G_i	The ID number of GM which node i belongs to
mG_i	The ID number of GM which node i is ready to join
BoolP_i	Denotes the state that node i must change another group to meet the group condition
Pointer S_i	Points to one of node i's neighbors which is invited to join the group which node i belongs to
Pointer L_i	Points to a certain node which node i accepts its invitation and agrees to join its group

$$D_{\min}(S) = i \in S \ \ when\ \forall j \in S, \begin{cases} \mathrm{Den}(i) < \mathrm{Den}(j) \\ \mathrm{Den}(i) = \mathrm{Den}(j)\ \text{if}\ i < j \end{cases}$$

Function 2: the maximum density set function $D_{\max}(S)$, where S is the input as well as the subset of set U, i.e., $S \in U$; the output of $D_{\max}(S)$ is one certain element of S satisfying the following equation:

$$D_{\max}(S) = i \in S\, when\ \forall j \in S, \begin{cases} \mathrm{Den}(i) > \mathrm{Den}(j) \\ \mathrm{Den}(i) = \mathrm{Den}(j)\ \text{if}\ i < j \end{cases}$$

Function 3: the group condition function $G_{\mathrm{con}}(i)$: In order to check whether the mentioned group conditions are satisfied around the neighbors of node i, the function $G_{\mathrm{con}}(i)$ is defined to output the smallest ID number of such eligible group's GM. If there are no eligible groups around the neighbors of node i, the function would return null. The detailed mathematical description is shown as follows:

$$G_{con}(i) = \begin{cases} j & if\left|\{u \in N_i^{open}, G_u = j\}\right| \geq \left|\{u \in N_i^{open}, G_u \neq j\}\right| \\ null & \text{if a group does not exist} \end{cases}$$

Function 4: the GM decision function GM(i) whose aim is to search the eligible group manager among node i and its neighbors. The input of GM(i) is the ID of node i and it outputs the ID of such eligible GM. If there is no appropriate GM, the function returns null. The pseudo-codes of function 4 are shown as follows:

***GM(i)*: The decision function for the GM attached to N_i^{open}**

Int*GM*(*i*)
{ **if** $(G_i = null \&\& D_{max}(N_i^{open}) == i)$
  return*i*;
if $(G_{con}(i) = null \&\& \forall j \in N_i^{open}, G_j \neq null)$
  return*i*;
if $(\exists j \in N_i^{open}, G_j == j)$
  return $(D_{max}(j \in N_i^{open}, G_j == j))$;
return null;
}

According to our group rules, for the strategy mobile nodes in MANETs, the group establishment will be completed when the condition $\forall i \in U,\ G_{\mathrm{con}}(i) = G_i \neq$ null is true.

3.1.3.3 Description of the group establishing algorithm

The pseudo-codes are as follows:

Group establishing algorithm

Initialization()
{**For**(i=1; i<=n; i++)
  { G_i = null; mG_i = null;
   $P_i = false$; $S_i = L_i$ =null;
   $G_{con}(i) = null$; }
}// Initialize and Clear the relative group variables buffers

Test1(*i*)
{
If $(\forall k \in N_i^{open}, S_k == i)$
  { $mG_i = G_k$; $L_i = k$; }
  Else If $(G_{con}(i) = null)$ $mG_i = GM(i)$;
  Else $mG_i = G_{con}(i) L_i = null$;
return;
} //Used for establishing of GM and joining group

Test2(*i*)
{
  If $((G_i \neq null) \&\& (a_i < Den(i) - a_i))$
  {
   $S_i = D_{min}(\{j | j \in N_i^{open}, G_j \neq G_i, L_j = null\})$;
   $mG_i = null$;
   If $(S_i = null)$ $mG_i = D_{max}(\{j | j \in N_i^{open}, G_j \neq G_i\})$;
  }
return;
} //Used for group users to recruit new member nodes

Test3 (*i*)
{
If $((mG_i \neq null) \&\& (G_i \neq mG_i))$ $P_i = true$;
return;
} //Used for a node when moving from a group to another group

Test4 (*i*)
{
If $(P_i == true)$
 {
If
  $((\forall j \in N_i^{open}, P_j == false) \| (\exists j \in N_i^{open}, (P_j == true) \| ((Den(i) < Den(j)) \| ((Den(i) == Den(j)) \&\& (i < j))))$
{
$G_i = mG_i$;
  }
  Else
  {
   $mG_i = null$; $S_i = null$; $P_i = false$;
  }
 }
return;
} //Force to assign the executing order for neighbor users which need to change groups simultaneously, i.e. execute changing operation from the user with smallest ID number.

Test5 (*i*)
{
If $(\exists k \in N_i^{open}, (G_i == k) \&\& (G_k \neq k))$
 {
   $G_i = mG_i = S_i = null$;
   $L_i = null$;

```
    }
    return;
}// Error processing condition, clear the relative group variables of node i when i indicates to an non-existent group

Test6 (i)
{
If (((L_i ≠ null)&&(G_{L_i} ≠ G_i))||(L_i ⊄ N_i^{open}))L_i = null;
return;
}       //

Check()
{
For(i=1; i<=n; i++)
    {
        Test1(i);
        Test2(i);
        Test3(i);
        Test4(i);
        Test5(i);
        Test6(i);
    }
}

void main()
{
Initialization();
While(End_Flag == 0)
    {
        For(i=1; i<=n; i++)
        {
            If (∀i ∈ U, G_con(i) == G_i ≠ null)
            {
                If(i==n)End_Flag = 1;
            }
    Else
            {
                Check();
                Break;
            }
        }
    }
}
```

For a specific circumstance, we take a network topology for instance to explain the process of our group establishing algorithm in detail. Briefly, there are 12 nodes implementing the strategy mobile application in MANETs. Each node runs the link quality detection algorithm to be aware of its neighbor table which contains members keeping up relatively stable correspondence with itself. Then members exchange IDs with each other. Finally, the real-time topology of MANETs is built as shown in Figure 1a.

The implementation of our algorithm can be generally divided into the following phases:

A. *Initialization phase*: Each node runs Den(i) to calculate its own density and exchanges this information with neighbors (as shown in Figure 1a, where one certain node is denoted by circles including its ID number, and its calculated density value is labeled nearby the circle). Firstly, the relative grouping variables and pointers of all nodes are set to null. Then each node is going to run through every test function (i.e., from Test1() to Test6()) by executing Check() until all the established groups are eligible under the defined rules.
B. *GM establishment and neighbor invitation phase*: After initialization, all nodes will accomplish the GM establishment by means of executing GM(i) when running through the Test1(). As shown in Figure 1b, node 3 and node 10 declare themselves GM for satisfying the defined GM condition. Then every GM executes Check() to traverse each test function for inviting enough neighbors to meet the group condition. For GM3, it runs Test2() and invites its neighbor whose density is the smallest (i.e., node 2) to join the group according to $S_i = D_{\min}\left(\{j | j \in \mathrm{N_i^{open}}, G_j \neq G_i, L_j = \mathrm{null}\}\right)$. Similarly, for GM10, its neighbors, nodes 8, 11, and 12, are invited into group 10 by multiple execution of Test2(). For nodes 4 and 5, which connect to different GMs (i.e., GM3 and GM10), they execute GM(i) when running through Test1() to choose the GM with the greatest density (see $D_{\max}\left(j \in \mathrm{N_i^{open}}, G_j = j\right)$) as their GM and join this group (i.e., nodes 4 and 5 join group 3). After all, GM3 and GM10 have invited and recruited enough neighbors according to $a_i \geq \mathrm{Den}(i) - a_i$, and this phase is finished.
C. *Exterior/edge member joining stage*: The remainder nodes which have not been invited (i.e., nodes 1, 6, 7, 9) execute Check() to run through each test function and choose the appropriate conditions to join the existing group. As shown in Figure 1c, we can see that node 1 joins group 3 by calculating the group condition function $G_{\mathrm{con}}(1) = 3$ and executing Test1() and Test3(). In the same way, nodes 6 and 7 also join group 3 by calculating $G_{\mathrm{con}}(6) = G_{con}(7) = 3$ and executing Test1() and Test3().
D. *Group completion stage*: In the above stage, for node 9, the group condition function satisfies $G_{\mathrm{con}}(9) =$ null when nodes 6 and 7 have not yet joined group 3. At this moment, by executing Test1(), it still cannot determine the affiliation of node 9 due to the result $G_9 = G_{\mathrm{con}}(9) =$ null. Therefore, in this phase, it will continue the ergodic operation to settle the comparable situation. As Figure 1d shows, after

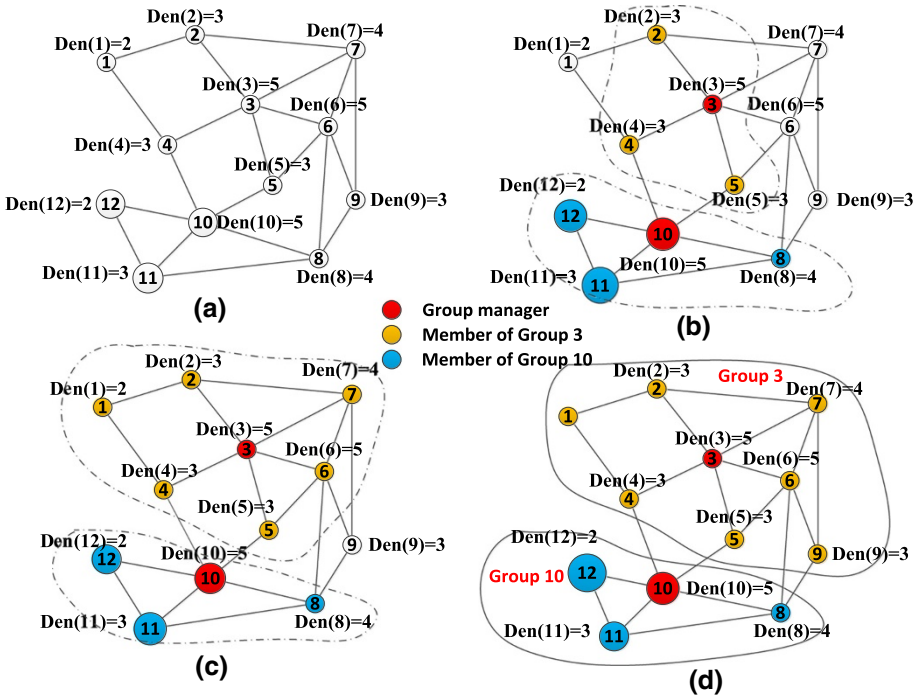

Figure 1 Grouping implementation. **(a)** Initialization phase. **(b)** GM establishment and neighbor invitation phase. **(c)** Exterior/edge member joining stage. **(d)** Group completion stage.

nodes 6 and 7 join group 3 through re-traversed operation, node 9 finally determines its affiliation (i.e., group 3). So far, all the members satisfy the terminal condition $\forall i \in U$, $G_{\text{con}}(i) = G_i \neq \text{null}$ and the groups for MANET secret communication are established successfully.

The above group establishing algorithm applied in MANETs with strategic mobile scenario has the following advantages:

1. The mobile nodes or users are divided into several communication groups in a self-organizing manner. It does not rely on a centralized server to deploy the relative grouping parameters in advance. What is more, each node only needs to interact with its own neighbors rather than maintain the information of other groups. Consequently, as the user number increases and the scale of group enlarges, each user still maintains the information and data processing within its own single and adjacent groups, which serve the needs of self-organization and scalability for MANETs.
2. According to the strategy mobile application scenario, when a user detects more high quality communication links, its movement state tends to be stable, i.e., the probability of large-scale stochastic move is relatively low. Under our group establishing algorithm, the user which has the greatest number of high-quality links is most likely to be a GM. As a matter of fact, GM needs to invite (or recruit) sufficient members to satisfy the group requirements designed for strategic mobile scenario; therefore, the majority of those members who share the high quality links with the GM could essentially be recruited to form a group. The probability of the GM and its neighbors of forming of the core region in one group and generating tremendous movement is relatively low. This advantage lays the foundation for reducing the energy consumption and communication overhead during the key updating and maintenance in MANETs.
3. Through executing the algorithm, the members in the edge region of one group are those with lower densities compared to those in the core region. As a result, the users in the edge region are most likely to generate the wide movement. We only pay more attention to implement the mobile management for users in the edge region of one group when designing our group key management protocol. Naturally, it will effectively reduce the resource consumption of design and maintenance for protocols.

3.2 Diffie-Hellman group key management

We adopt the Diffie-Hellman (DH) problem as the way of key exchange. Let G want be a finite multiplicative

group of some large prime order q where the famous-known discrete logarithm (DL) problem is believed to be intractable, and let g be a generator of G. Also consider a hash function H: $\{0, 1\}^* \rightarrow Z_q^*$.

Assume that multimembers are denoted by set $\{N_i | \ i = 1, 2,\ldots, n\}$, and GM wants to negotiate a shared key. They perform the following steps:

- Each member N_i chooses a random private ephemeral key $r_i \in Z_q^*$ and calculates the blind version g^{ri} to GM.
- The GM raises each received version to its own private ephemeral key r_m and broadcasts them along with the original contributions to the group, i.e., it sends $\{g^{ri}, g^{ri\ rm}\}$ for all $i = 1, 2,\ldots, n$.

Then each member i checks if its contribution is included correctly, removes its private ephemeral key for $i = \{1, 2,\ldots, n\}$ to get g^{rm}, and computes the shared key of the group as shown by the following:

$$K_s = g^{r_m} \cdot \prod_{i=1}^{n} g^{r_i r_m} = g^{r_m\left(1+\sum_{i=1}^{n} r_i\right)}$$

3.3 Strategic mobile management

When designing an appropriate key management protocol for MANETs, it should be well considered for the following two aspects. On the one hand, it needs to adopt the approach of distributed group key management, i.e., the users are divided into multiple groups and the procedures of key establishment, distribution, and maintenance are executed only within each group. On the other hand, the dynamics of MANET topology and node mobility must be considered to put rekeying and maintaining into effect. Therefore, the design of a highly efficient mobile management among users plays a significantly important role in cutting down the cost of group rekeying, improving the scalability and efficiency of key management.

In the strategic mobile scenario of MANETs, since nodes in the edge region are more likely to move than those in the core region, we mainly focus on investigating the mobile management method for nodes in edge regions. The ID numbers of mobile nodes in a certain edge region which move to another group are coded and decoded by using the Prufer codec method. And they are managed by the involved GMs. Thus, it is unnecessary to restart the rekeying process once nodes in the edge region of a group move to other adjacent groups. It only requires the involved GMs to manage the mobility of roaming nodes, which can carry out high efficiency of key management for MANETs.

4 Group key management

In this section, we mainly present our group key management for the strategic mobile scenario of MANETs which relies on the aforementioned grouping algorithm and DH method. Notice that each node is pre-loaded a unique ID number and a uniform Hash function in advance.

4.1 Group architecture and establishment

Through running the group establishing algorithm designed in the previous section, all users in MANETs are divided into multiple groups. Based on them, shown in Figure 2, we first describe the group architecture in this section.

Each communication group is composed of a GM and several group members $M_i (i = 1, 2,\cdots, n)$. The group members with one direct hop link to GM are defined as group backbone nodes (GBN). The member nodes which simultaneously bridge two or more GMs with one hop are called network bridges (NBs). Therefore, those groups connected by NBs are defined as adjacent groups. Note the transitivity property of adjacency relationship in our group architecture, i.e., if group G10 is adjacent to G3, meanwhile G3 is adjacent to G13, and then G10 and G13 are still regarded as adjacent groups. The set of all groups maintaining the same adjacent relationship is defined as adjacency field (AF) in our group architecture. Different AFs are distinguished by different subscript numbers, such as AF_1, AF_2, and so on.

There are three kinds of group keys in our proposed key management protocol. They are group session key K_{G_i}, adjacency field key K_{AF}, and mobile management key $K_{\mathrm{mob}}^{\mathrm{AF}}$. Specifically, K_{G_i}, shared by the GM and its member nodes, is used for secret information broadcast and multihop transmission in this group. K_{AF} is shared by all GMs and NBs residing in one adjacency field. It is used for adjacency information interaction and mobile management. $K_{\mathrm{mob}}^{\mathrm{AF}}$ is jointly generated by all GMs in the same AF and distributed to all nodes in this AF. $K_{\mathrm{mob}}^{\mathrm{AF}}$ is used for roaming nodes to transmit secret information between the host group and original group.

4.2 Group Diffie-Hellman (GDH) key management

4.2.1 Key establishment

4.2.1.1 Establishment of group session key (K_{G_i}) As shown in Figure 3, group session key is calculated by contributive agreement approaches such as GDH, which means all member nodes in the same group contribute to the computation of group session key. This protocol needs only two communication rounds to compute K_{G_i}

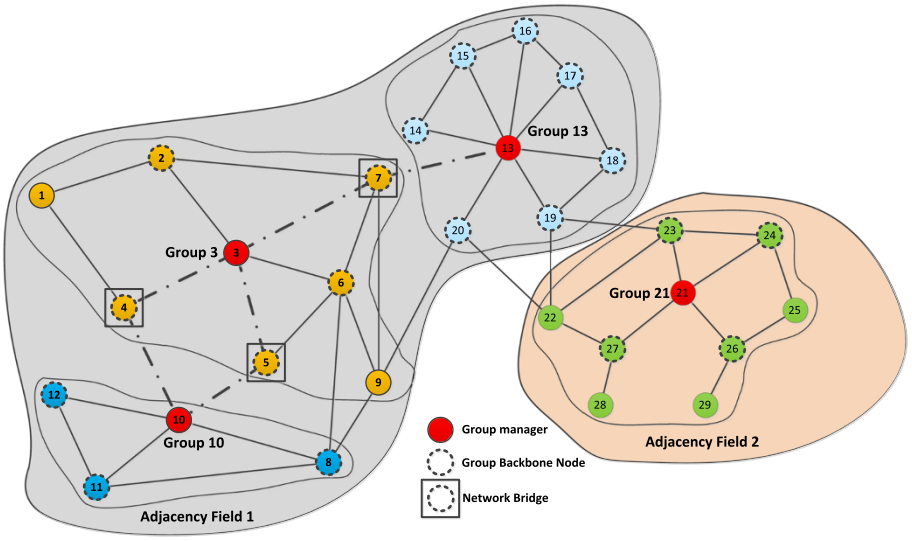

Figure 2 Group architecture.

after authentication. The members of the group execute the following two steps:

Step1: Each member M_i of group G_s generates a random number $r_i \in Z_q^*$ and calculates the version $\{N_i = g^{r_i} \bmod n\}$. Then M_i sends N_i to its group manager M_{G_i} as follows:

$$M_i \rightarrow M_{G_i} : \left\{\mathrm{ID}_{M_i} \| \mathrm{ID}_{M_{G_i}} \| N_i\right\} \| \mathrm{MAC}\left\{\mathrm{ID}_{M_i} \| \mathrm{ID}_{M_{G_i}} \| N_i\right\}$$

Step2: The group manager M_{G_i} authenticates the ID of M_i and raises the received N_i generated by M_i to its own random version $\{N_c = g^{r_c} \bmod n\}$. Then M_{G_i} broadcasts them along with the original contributions to the group, for example, it sends the new blinded factor $\left\{N_i^c = (N_i)^{r_c} \bmod n = g^{r_c r_i} \bmod n\right\}$ to M_i as given by:

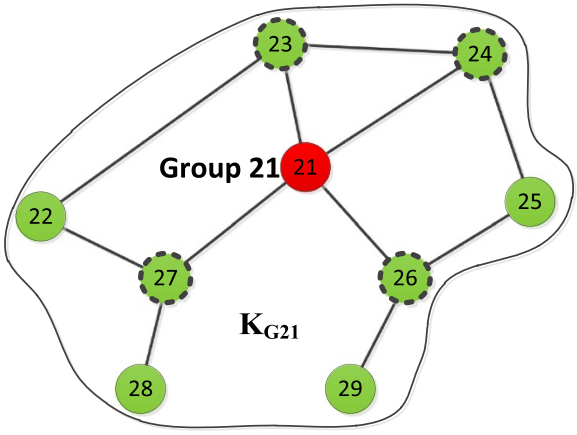

Figure 3 Establishment of group session key.

$$M_{G_i} \rightarrow M_i : \left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{M_i} \| N_i \| N_i^c\right\} \| \mathrm{MAC}\left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{M_i} \| N_i \| N_i^c\right\}$$

Then each M_i checks if its contribution is included correctly. After authenticating the packet, M_i removes g^{r_i} from $\left\{N_i^c = (N_i)^{r_c} \bmod n = g^{r_c r_i} \bmod n\right\}$ to obtain g^{r_c}. All member nodes of group manager M_{G_i} compute the shared key K_{G_i} of this group as shown by following:

$$K_{G_i} = g^{r_c} \cdot \prod_{i=1}^{n} g^{r_i r_c} = g^{r_c\left(1+\sum_{i=1}^{n} r_i\right)}$$

4.2.1.2 Establishment of adjacency field key (K_{AF})

Once all of the group session keys in a certain AF are established, then the establishment of adjacency field key can be generated. The establishment of K_{AF} is launched by the GM which owns most NBs in one AF. If there are GMs which have the same number of NBs, the GM with the smallest ID number will be in charge for launching the K_{AF} establishment procedure. As shown in Figure 4, node 3 is in charge of the establishment of K_{AF1}.

Step 1: The GM which is responsible for the establishment of K_{AF} (adjacency field manager (AFM)) sends the key establishment request (AF key establishment request) to all the NBs of its neighbors, and the NBs forward this request to the rest of the GMs of the adjacent groups. The GMs of the adjacent groups which receive the request forward it to the rest of the NBs to spread until all the GMs in the AF receive the request.

Step 2: Each group manager M_{G_i} which received the establishment request of this adjacency field generates a random number $r_i^{\mathrm{AF}} \in Z_q^*$ and calculates the version

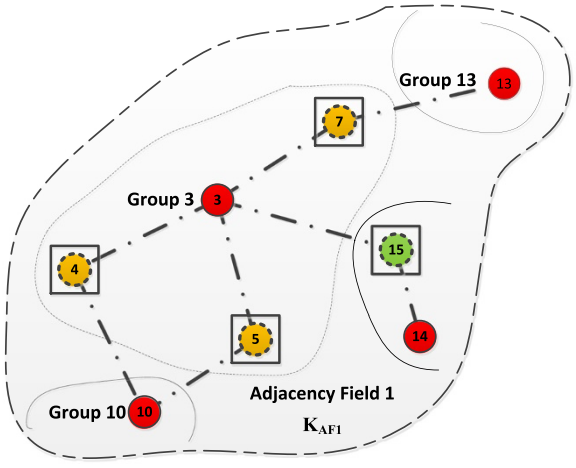

Figure 4 Establishment of adjacency field key.

$\left\{N_i^{\mathrm{AF}} = g^{r_i^{\mathrm{AF}}} \bmod n\right\}$. Then M_{G_i} sends N_i^{AF} to its AFM M_{AFM_i} through the collaborative forwarding among this field's NBs as given by:

$$M_{G_i} \rightarrow \mathrm{NB} \rightarrow M_{AFM_i} : \left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{AFM_i} \| N_i^{AF}\right\} \| \mathrm{MAC}\left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{AFM_i} \| N_i^{AF}\right\}$$

Step 3: The AFM M_{AFM_i} authenticates the ID of M_{G_i} and raises the received N_i^{AF} generated by M_{G_i} to its own random version $\left\{N_c^{AF} = g^{r_c^{AF}} \bmod n\right\}$. Then M_{AFM_i} broadcasts the raised result along with the original contributions to the group manager, for example, it sends the new blinded factor $\left\{N_{i|c}^{\mathrm{AF}} = \left(N_i^{AF}\right)^{r_c^{\mathrm{AF}}} \bmod n = g^{r_i^{AF} r_c^{AF}} \bmod n\right\}$ to M_{G_i} through the forwarding among this field's NBs as given by:

$$M_{\mathrm{AFM}_i} \rightarrow \mathrm{NBs} \rightarrow M_{G_i} :$$

$$\left\{\mathrm{ID}_{\mathrm{AFM}_i} \| \mathrm{ID}_{M_{G_i}} \| N_i^{AF} \| N_{i|c}^{AF}\right\} \| \mathrm{MAC}\left\{\mathrm{ID}_{\mathrm{AFM}_i} \| \mathrm{ID}_{M_{G_i}} \| N_i^{AF} \| N_{i|c}^{AF}\right\}.$$

Then each M_{G_i} checks if its contribution is included correctly. After authenticating the packet, M_{G_i} removes $g^{r_i^{\mathrm{AF}}}$ from $\left\{N_{i|c}^{\mathrm{AF}} = \left(N_i^{\mathrm{AF}}\right)^{r_c^{\mathrm{AF}}} \bmod n = g^{r_i^{\mathrm{AF}} r_c^{\mathrm{AF}}} \bmod n\right\}$ to obtain $g^{r_c^{\mathrm{AF}}}$. All GMs M_{G_i} as well as AFM M_{AFM_i} in this AF compute the shared AF key K_{AF_i} as shown by the following:

$$K_{\mathrm{AF}_i} = g^{r_c^{\mathrm{AF}}} \cdot \prod_{i \in \mathrm{AF}} g^{r_i^{\mathrm{AF}} r_c^{\mathrm{AF}}} = g^{r_c^{\mathrm{AF}}} \left(1 + \sum_{i=1}^{n} r_i^{\mathrm{AF}}\right)$$

4.2.1.3 Establishment of mobile management key ($K_{\mathrm{mob}}^{\mathrm{AF}}$)

In our key management protocol, when a node of low density roams from the original group to its adjacent group in order to detect more effective links, our objective is to realize the secure communication between roaming node and its original group without updating the group session key. To this end, it mainly depends on the mobile management using $K_{\mathrm{mob}}^{\mathrm{AF}}$.

After establishing K_{AF_i} of one certain AF_i, all GMs in AF_i can directly calculate by a hash function.

Step 1: The adjacency field manager M_{AFM_i} generates an initial value f of a fresh counter and sends it to all group mangers M_{G_i} of this adjacency field encrypted by K_{AF_i} through the forwarding among this field's NBs as follows:

$$M_{\mathrm{AFM}_i} \rightarrow \mathrm{NBs} \rightarrow M_{G_i} :$$

$$E_{K_{\mathrm{AF}_i}}\left\{\mathrm{ID}_{\mathrm{AFM}_i} \| \mathrm{ID}_{M_{G_i}} \| f\right\} \| MAC\left(E_{K_{\mathrm{AF}_i}}\left\{\mathrm{ID}_{\mathrm{AFM}_i} \| \mathrm{ID}_{M_{G_i}} \| f\right\}\right)$$

Step 2: The group manager M_{G_i} decrypts the packet and checks if both of the IDs of M_{AFM_i} and M_{G_i} are embedded correctly. After authenticating the packet, the group manager M_{G_i} obtains f and computes the mobile management key $K_{\mathrm{mob}}^{\mathrm{AF}i} = \mathrm{MAC}(K_{\mathrm{AF}_i}, f)$ of the adjacency field and disseminates it to all the members of its own group encrypted by group session key K_{G_i} as follows:

$$M_{G_i} \rightarrow M_i$$

$$E_{K_{G_i}}\left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{M_i} \| K_{\mathrm{mob}}^{\mathrm{AF}i}\right\} \| \mathrm{MAC}\left(E_{K_{G_i}}\left\{\mathrm{ID}_{M_{G_i}} \| \mathrm{ID}_{M_i} \| K_{\mathrm{mob}}^{\mathrm{AF}_i}\right\}\right)$$

Then every member of the adjacency field could share the mobile management key $K_{\mathrm{mob}}^{\mathrm{AF}_i}$ securely.

4.2.2 Group key updating

In traditional group key management protocols for MANETs, it often requires frequent key updating for the purpose of guaranteeing the security of key protocol, which primarily refers to the forward and backward security and the ability against malicious attacks. Even though taking periodical rekeying scheme, once group topology changes due to the roam of members, it will launch the extra key updating procedure. What is more, consider the following situations: 1) Some legal nodes cannot be detected by their companions in one group because of the dramatic drop of the surrounding communication quality in just a very short time. 2) Some legal nodes roam randomly in a certain range which results for nodes to leave the group at some moment but return to the group at the next moment, etc.

The above two cases are deemed as group topology changes. Once they happen, the GM launches the group key updating process and distributes the new shared key to all the remaining members. If these leaving members return to the original group after a short period of time, they would be treated as new members joining the group. Therefore, the GM will launch the group key

updating process again. However, in MANETs when a member which roams back and forth across the group border, or is out of touch not due to its hardware fault, is kind of a legal node, it is unnecessary to waste lots of communication and computation overhead for rekeying in the abovementioned situations.

In order to reduce the frequency of the unnecessary key updating in the strategic mobile scenario of MANETs, our proposed protocol divides the MANETs into the core region and edge region. The core region of a group consists of members with high density and the backbone nodes (backbone nodes are the collection of nodes with higher density and own the dominant number of high-quality communication link inside this group), while the edge region of the group is composed of members with relatively lower density which may generate wide movements with a larger probability. In addition, we define the adjacency field cascading two or more groups with more interactive links. To this end, on the one hand, users with more communication links and lower mobile probability are gathered together to reduce the possibility of rekeying in one group. On the other hand, the mobile management model, described in detail in Section 4.2.3, for users in edge regions of one AF ensure that the nodes will still be able to realize confidential communication with the original group without updating the session key when they depart from the original group but still in the adjacency region. To sum up, in our protocol, the corresponding group key updating procedure will be triggered when the following situations occur:

1. It will perform the key updating process once the default updating cycle is triggered.
2. When mobile nodes voluntarily leave the original group to join the new group (i.e., sending a leaving request) or when the mobile nodes roam to the outside of their adjacent field and are detected in other AFs, which leads to the essential changes of the group members, it will perform the key updating process.
3. When a node fails and cannot be repaired within a certain time, or when the edge region nodes (or backbone nodes) roam to the outside of their adjacent field, the GMs will not be able to locate the roaming nodes in the AF by means of the mobile management model. Thereby, it will perform the key updating process.
4. When the large-scale movement happens to the GMs or their backbone members, which causes the disruption of the group structure, the group establishing algorithm would be re-executed to construct new group architectures. As a consequence, various types of keys in our protocol will be updated after completing the grouping procedure.

4.2.3 Secure communication among roaming nodes

4.2.3.1 Strategic mobile management in adjacency field In this section, we will describe our strategic mobile management model within the adjacency field. The goal of this model is to provide the relay services and mobile management for roaming nodes and the original GM to realize the confidential communication, which is achieved by multibroadcasting technology using a position table built and maintained by all GMs and NBs in the adjacency field. The model is described in detail as follows:

A. When the member M_i^{Ori} (probable in the edge region of the original group) of a group takes the strategy mobile scheme to move into the other group but still in its adjacent field, it generates the following packet (Hello packet) to detect links and interact with members of a new group represented as M_j^{NewG} for instance:

$$M_i^{OriG} \rightarrow M_j^{NewG}$$

$$Hello \| E_{K_{mob}^{\mathrm{AF}s}} \left\{ \mathrm{ID}_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_i^{OriG}} \right\} \| \mathrm{MAC}\left(\mathrm{Hello} \| E_{K_{\mathrm{mob}}^{\mathrm{AF}s}} \left\{ ID_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_i^{\mathrm{OriG}}} \right\} \right)$$

where $\mathrm{ID}_{M_{\mathrm{OriG}_i}}$ is the ID number of the original group manager and $\mathrm{ID}_{M_i^{\mathrm{OriG}}}$ is the ID number of the mobile node. The Hello packet is encrypted with adjacent field mobile management key.

B. After the new group member M_j^{NewG} receives the Hello packet, it decrypts the packet with $K_{\mathrm{mob}}^{\mathrm{AF}s}$ to obtain $\mathrm{ID}_{M_{\mathrm{OriG}_i}}$ and $ID_{\mathrm{M_i}^{OriG}}$. Then M_j^{NewG} sends an ACK to inform M_i^{Ori} the ID number of this new group manager, which is shown as follows:

$$M_j^{\mathrm{NewG}} \rightarrow M_i^{\mathrm{OriG}}$$

$$\mathrm{ACK} \| E_{K_{\mathrm{mob}}^{\mathrm{AF}s}} \left\{ \mathrm{ID}_{M_{\mathrm{NewG}_j}} \| \mathrm{ID}_{M_j^{\mathrm{NewG}}} \right\} \| \mathrm{MAC}\left(\mathrm{ACK} \| E_{K_{\mathrm{mob}}^{AF_s}} \left\{ \mathrm{ID}_{M_{\mathrm{NewG}_j}} \| \mathrm{ID}_{M_j^{\mathrm{NewG}}} \right\} \right)$$

If the number of the ACK received from the new group by M_i^{Ori} is greater than the link number of the original group, M_i^{Ori} will reside in the new group at a larger probability. To this effect, M_i^{Ori} sends the resident request to M_j^{NewG} as follows (residing does not mean

joining; hence, M_i^{Ori} cannot share the session key with the members of the host group):

$$M_i^{\mathrm{OriG}} \rightarrow M_j^{\mathrm{NewG}}$$

$$\mathrm{RR} \| E_{K_{\mathrm{mob}}^{\mathrm{AF_s}}} \left\{ \mathrm{ID}_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_{\mathrm{NewG}_j}} \right\} \| \mathrm{MAC}\left(\mathrm{RR} \| E_{K_{\mathrm{mob}}^{\mathrm{AF_s}}} \left\{ \mathrm{ID}_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_{\mathrm{NewG}_j}} \right\} \right)$$

where $\mathrm{ID}_{M_{\mathrm{NewG}_i}}$ is the ID number of the new group manager.

C. After receiving the reside request sent by M_i^{Ori}, M_j^{NewG} decrypts it with $K_{\mathrm{mob}}^{\mathrm{AFs}}$ and verifies the involved ID numbers. Once they are authenticated successfully, M_j^{NewG} sends the information of the resident mobile node as well as its group information encrypted with the session key of the new group to GM (if it does not communicate directly with the GM, the packet can be forwarded to the GM by intermediate members), which is shown as follows:

$$M_j^{\mathrm{NewG}} \rightarrow M_{\mathrm{NewG}_j}:$$

$$\mathrm{Mob} \| E_{K_{\mathrm{NewG}_j}} \left\{ \mathrm{ID}_{M_j^{\mathrm{NewG}}} \| \mathrm{ID}_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_i^{\mathrm{OriG}}} \right\} \| \mathrm{MAC}\left(\mathrm{Mob} \| E_{K_{\mathrm{NewG}_j}} \left\{ \mathrm{ID}_{M_j^{\mathrm{NewG}}} \| \mathrm{ID}_{M_{\mathrm{OriG}_i}} \| \mathrm{ID}_{M_i^{\mathrm{OriG}}} \right\} \right)$$

where $\mathrm{ID}_{M_j^{\mathrm{NewG}}}$ is the ID number of the new group member which receives the Hello packet.

D. After executing the above three steps, all roaming nodes residing in the new groups of the same AF establish the relationship with the GMs of the new groups. All group managers in this AF share and maintain a position table storing each resident roaming node's location through spreading it by NBs (in this case, it requires to encrypt the information of the position table using the adjacent field key). If the resident information changes, it has to update the position table in time so as to realize the strategy mobile management within the AF.

4.2.3.2 Construction of secure multicast path among roaming nodes using the Prufer codec method If the GM needs to launch secret sessions within its group members, in addition to communicating with members still staying in the group, the GM has to search for roaming members residing in another group of the AF through the mobile management model described in the previous section. Then the GM could transmit confidential information to those members by means of relay forwarding responded by the involved GMs and NBs.

A traditional method of routing messages to objective roaming members is the way of flooding. However, the implementation of this method not only consumes too much intermediate nodes' energy but also generates a great number of redundant packets which results in consuming lots of bandwidth and inducing delay.

In order to launch the secure communication to target host group managers, a communication-efficient multicast path is required. In our scheme, we present a multicast path construction scheme based on the Prufer codec method. Using the Prufer codec method, we can significantly reduce the multicast's complexities in terms of both communication overhead and storage overhead. Specifically, a secure communication path calculated by using the topology-aware routing protocols for *ad hoc* [29,30] is created and encoded to the Prufer sequence. This secure communication path, from the original group manager to the target group manager, can be obtained by computing a minimum cost multicast tree, commonly known as the Steiner tree. Then we convert this tree into the Prufer sequence and embed it in the confidential packet for multicast, which is equivalent to multicasting the packet to multiple targets over a multihop network. Hence, this packet can be transmitted purposefully to the target group according to the Prufer sequence along the tree path instead of flooding diffusion.

In this section, the Prufer codec algorithm will be introduced first, i.e., how to convert a tree into the sequence information (encoding) and how to recover this unique tree by means of sequence transforming (decoding). Then we will illustrate a secret session instance between the roaming member and original group by using our key management protocol and secure communication codec-based multicast.

4.2.3.2.1 The Prufer codec method Suppose that there is a tree T denoted as $\{v_1, v_2, \cdots, v_n\}$, where $v_i (i = 1, 2, \cdots)$ represents the vertex of this tree (i.e., the MANET nodes in the tree). The value of v_i indicates the ID number of the node. Edge(v_i,v_j) is the link that connects nodes v_i and v_j.

1. The Prufer encoding algorithm
   - If v_i is a leaf node of tree T with the smallest ID number and v_j directly connects v_i, i.e., Edge(v_i,v_j) exists. Then put v_j into the Prufer sequence represented as PS.
   - Remove v_i and Edge(v_i, v_j) from tree T.
   - Go back to step 1 and re-execute the steps until only one edge remains in tree T. Then the sequence PS has been built successfully and put the remaining nodes of tree T into the other sequence represented as RPS (remaining of the Prufer sequence) in ascending order by their ID number.
2. The Prufer decoding algorithm

- Take the first element from RPS and PS, respectively (assume that they are v_i and v_j) and recover an edge Edge(v_i,v_j) of tree T.
- Remove v_i and v_j from RPS and PS, respectively. After that if v_j does not exist in PS, then put it into RPS with a proper place (arrange the ID number in ascending order).
- Go back to step 1 and re-execute the steps until there is no element in PS. At this moment, there are two remaining elements in RPS. Then recover the edge composed of these two remainder elements of tree T. So far, the whole tree T is successfully restored from PS and RPS.

4.2.3.2.2 Communication instance Assume a typical application in MANETs with strategic mobile scenario. Firstly, running the group establishing algorithm, users are divided into multiple adjacency fields and eligible groups according to our group rules. After that, the session key, adjacency field key, and mobile management key are established with a fewer rounds and lower latency through the previous key establishing procedure.

Take a certain adjacency field as an example represented in Figure 5a; there are five groups in AF_1, where nodes marked yellow are members of group 3. Due to strategy mobile, members 1, 2, and 8 of group 3 roam to other groups and detect more links. They choose to reside in group 6, group 13, and group 10, respectively, in AF_1. GM3 together with GM6, GM11 and GM13 of AF_1 creates and maintains a common position table which contains the above resident information by means of the mobile management model. If GM3 wants to communicate with its member nodes, it will query this table to get the resident information of its members in AF_1. Getting back to this instance, it will learn that group 6, group 13, and group 10 are the target groups. In addition,

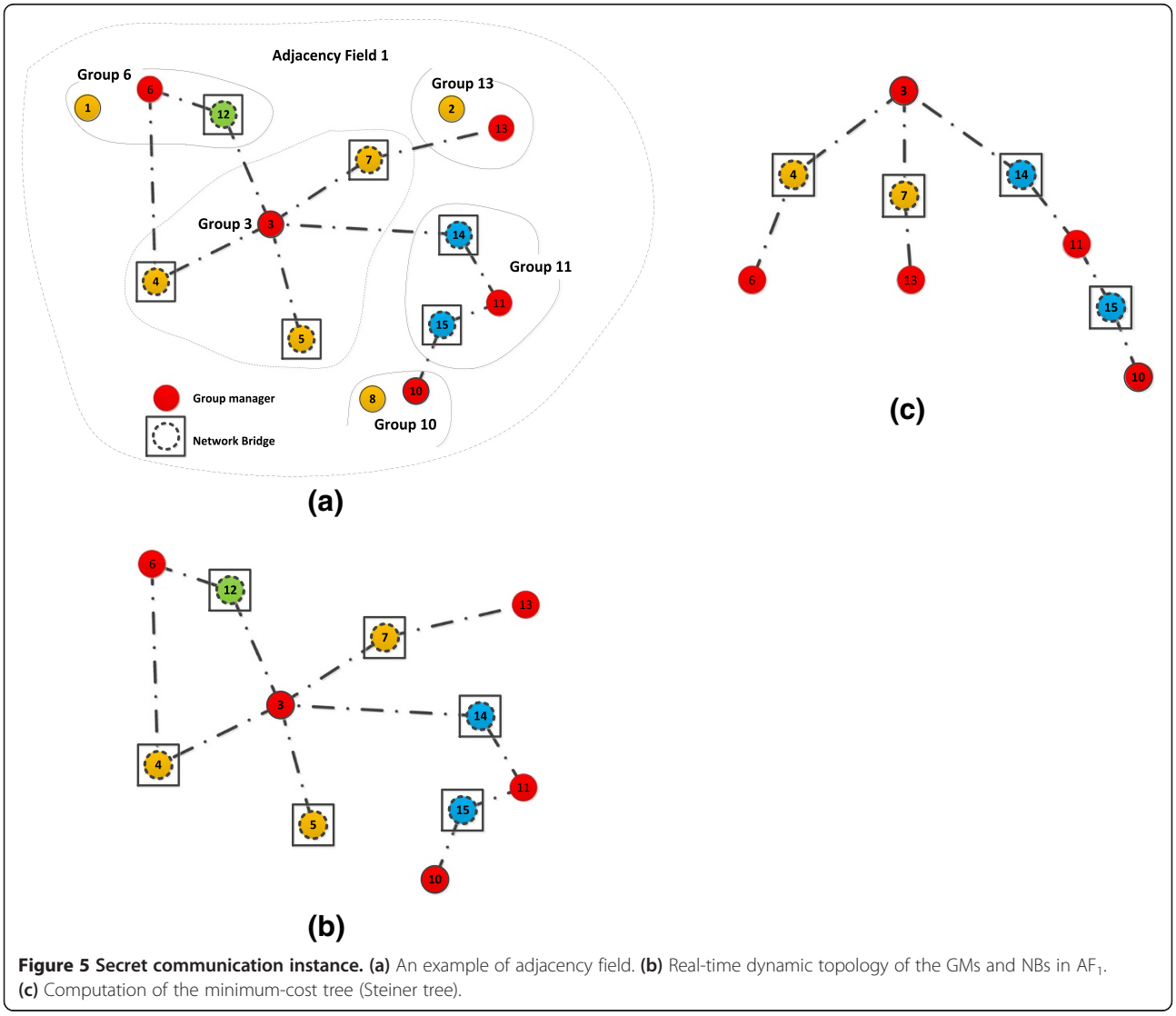

Figure 5 Secret communication instance. (a) An example of adjacency field. **(b)** Real-time dynamic topology of the GMs and NBs in AF_1. **(c)** Computation of the minimum-cost tree (Steiner tree).

through the topology-aware *ad hoc* routing protocols, the real-time dynamic topology of the GMs and NBs in AF_1 is perceived as shown in Figure 5b. Next, according to the target groups {6, 13, 10}, using the method [31,32], the minimum-cost tree (Steiner tree) can be computed, which is shown in Figure 5c. Finally, GM 3 converts this tree into sequence by the Prufer codec method and embeds it into multicast packet transmitting to the target users.

We describe the Prufer encoding procedure in detail as follows: Firstly, node 6 is the leaf node with the smallest ID number. Since Edge(6,4) is an edge of the tree, put node 4 into to PS (PS = {4}). Then node 6 and Edge (6,4) are removed from the tree. Secondly, node 4 becomes the leaf node with the smallest ID number and Edge(4,3) is also an edge. Thus, put node 3 into PS (PS = {4,3}) and remove node 4 and Edge(4,3) from the tree. Similarly, the above process is repeated until there is only one edge in the tree. Finally, the two sequences, PS = {4,3,15,7,3,14,11} and RPS = {6,10,13} are computed successfully. After that, GM3 generates the following multicast packet encrypted by K_{AF_1} and sends it to NBs 4, 7, and 14:

$$M_{G_3} \rightarrow \begin{cases} M_4^{\mathrm{NB}_3} \\ M_7^{\mathrm{NB}_3} \\ M_{14}^{\mathrm{NB}_{11}} \end{cases}$$

$$\mathrm{Multicast} \| E_{K_{\mathrm{AF1}}} \left\{ \mathrm{ID}_{M_{G_3}} \| E_{K_{\mathrm{mob}}^{\mathrm{AF}_1}} \{ \mathrm{PS} \| \mathrm{RPS} \} \right\} \| E_{K_{G_3}} \{ ''\mathrm{Group\ Data}'' \}$$

$$\| \mathrm{MAC} \Big(\mathrm{Multicast} \| E_{K_{AF1}} \left\{ ID_{M_{G_3}} \| E_{K_{\mathrm{mob}}^{AF_1}} \{ \mathrm{PSRPS} \} \right\} \| E_{K_{G_3}} \{ ''\mathrm{Group\ Data}'' \} \Big)$$

where the multicast packet includes multicast frame header, PS, and RPS encrypted with $K_{\mathrm{mob}}^{\mathrm{AF}_1}$ and the confidential original group data encrypted with the session key of group 3, i.e., K_{G_3}.

After the involved NBs receive the multicast packets, they decrypt the packet by $K_{\mathrm{mob}}^{\mathrm{AF}_1}$ to obtain PS and RPS, and it will restore the minimum-cost tree through the Prufer decoding algorithm as follows: Firstly, take out the first element from RPS (i.e., node 6) and PS (i.e., node 4) to recover an edge Edge(6,4) of the tree. Then remove node 4 from RPS, and similarly remove node 6 from PS. Since node 4 does not exist in PS, put node 4 into RPS with a proper place. At this moment, PS equals {3,15,7,3,14,11} and RPS equals {4,10,13}. Secondly, node 4 and node 3 are taken out, respectively, form RPS and PS to recover an edge Edge(4,3) of the tree and then they are removed from corresponding sequences. As node 3 still exists in PS, it will not be put into the RPS. Right now, PS equals {15,7,3,14,11} and RPS equals {10,13}. Similarly, in the next phase, Edge (10,15) is recovered, and PS changes to {7,3,14,11} and RPS changes to {13,15}. With the continuous implementation of the decoding algorithm, Edge(13,7), Edge(7,3), Edge(3,14), and Edge(14,11) are successively recovered. As a consequence, there is no element in PS and RPS that equals {11,15}. Finally, these two elements in RPS are taken out to recover the last edge Edge(15,11) of the tree. So far, the minimum tree is restored successfully. The different nodes of the tree take different measures to the multicast packet according to the following rules:

1. If a node, which is included in the initial PS but not in the target group (i.e., {6, 13, 10}), receives the multicast packet (i.e., 4, 3), it should forward the packet directly.
2. If a node, which is neither in the target group {6, 13, 10} nor in the initial PS, receives the multicast packet (i.e., 5, 12), it should discard the packet directly.
3. If a node, which does not belong to the initial PS but belongs to the target ({6, 13, 10}) receives the multicast packet, it should process the packet as follows: It first extracts $E_{K_{G_3}}\{'\mathrm{Group\ Data}'\}$ and $\mathrm{ID}_{M_{G_3}}$ from the multicast packet, then builds a new secret packet as follows. The packet is encrypted by $K_{\mathrm{mob}}^{\mathrm{AF}_1}$ and sent to the target roaming nodes (in this example, node 6 processes the multicast packet and sends the secret packet to node 1 in group 6).

$$M_{G_3} \rightarrow M_1^{\mathrm{OriG}_3}:$$

$$\mathrm{SecCom} \| E_{K_{\mathrm{mob}}^{\mathrm{AF}_1}} \left\{ \mathrm{ID}_{M_{G_3}} \| E_{K_{G_3}} \{ ''\mathrm{Group\ Data}'' \} \right\}$$

$$\| \mathrm{MAC} \left(\mathrm{SecCom} \| E_{K_{\mathrm{mob}}^{\mathrm{AF}_1}} \left\{ \mathrm{ID}_{M_{G_3}} \| E_{K_{G_3}} \{ ''\mathrm{Group\ Data}'' \} \right\} \right)$$

In the above communication instance, without updating the original group session key, the members generating strategy roaming scheme are still able to maintain confidential communication with the original group in the adjacent field by means of mobile management and efficient multicast. It can effectively reduce the energy cost and improve the efficiency of multicast, which is suitable for MANET environment, especially for the MANETs under the strategic mobile scenario.

5 Performance analyses

5.1 Attack model and security countermeasures

Generally speaking, an attacker would strive to obtain authorized access and then implement as an internal attacker. First, we briefly introduce feasible external attack models and our countermeasures by using group key management. An external attacker can obtain unauthorized access to a legitimate field by eavesdropping packets or any message embedding various keys for more advanced attacks. We adopt independent rekeying at three different

levels (i.e., group session key, adjacency field key, and mobile management key) to guarantee the network confidentiality. An external attacker can also try to modify the packets to break their integrity. We use a contributory key (the group key) shared by only legitimate group members to guarantee the data integrity. An external attacker might pretend a legitimate group member to join a group. We have each node pre-loaded with a hash function to ensure every message's authenticity by message authentication code (MAC) and prevent potential impersonation attacks during the authentication process of a new member's joining. Active external attacks such as denial-of-service (DoS) attacks can also be mitigated by MAC. An external attacker may forge messages. Since only legitimate group members with a related session key distributed can comprehend messages communicated by other group members, forged messages will certainly be discarded.

5.2 Simulation

To verify and explore our design, we have used three powerful simulators including MATLAB, NS2, and OPNET for evaluating the group establishing algorithm and group key management. All of these simulators are having lots of libraries for MANETs.

5.2.1 Computational complexity and convergence analyses for the group establishing algorithm

In this section, we mainly focus on estimating the computational complexity and convergence for our group establishing algorithm. According to the algorithm, the procedure of running through the six condition functions continues if the value of $G_i(i = 1,2,\ldots, n)$ of certain nodes is still changing. The immediate cause of changing G_i is to meet the executive requirement of Test4() as described in the previous section. Thus, the executive times of Test4() can be used for estimating the computational complexity of our group establishing algorithm.

Theorem 1: There are n nodes randomly deployed in a MANET region, and the value of maximum density among all nodes is Δ. By means of our group establishing algorithm, starting from an arbitrary illegitimate group state, Test4() would be executed at most $n(\Delta + 1)$ times to complete the grouping procedure.

Proof: When an arbitrary member M_i checks the six test functions, if $M_i \neq G_{\text{con}}(i)$, through computing mG_i first, then M_i will execute Test4() to let G_i point to $G_{\text{con}}(i)$. Similarly, if $M_i = G_{\text{con}}(i)$, M_i may also execute Test4() to handle the invitation generated by neighbors of M_i which have not satisfied group conditions. To sum up, according to our algorithm, M_i will not execute Test4() to change G_i again when all its neighbors M_j meet the condition $M_j = G_{\text{con}}(j)$. Thus, in the worst case, M_i will execute Test4() Δ times. There are n nodes in the network; therefore, Test4() would be executed at most $n(\Delta + 1)$ times to finish the grouping procedure.

Theorem 2: The group establishing algorithm converges in $O(\Delta^2 n^2)$ time steps

Proof: According to Function4(), if there are several neighbors of M_i, including itself, they must execute the group changing operation simultaneously, and the member with the smallest ID number will execute the operation first. Then the remaining members have to re-estimate the variables mG_i and $G_{\text{con}}(i)$ by running through the involved condition functions. For M_i, the worst case is that the member with the largest ID number which is ready to perform changing operation is M_i; thus, it will re-estimate $G_{\text{con}}(i)$ Δ times. Between two consecutive executions, there could be at most Δ time steps on M_i. Therefore, for all MANETs, the group establishing algorithm converges in $O(\Delta \times n(\Delta + 1) \times n) = O(\Delta^2 n^2)$ time steps.

The above analyses can indicate that our group establishing algorithm meets the requirement of computational complexity and convergence for MANETs.

Next, to compare the grouping performance of our algorithm with typical grouping algorithms commonly known as MobHid [23], LLACA [14,25], and DLA-DC [24], we have conducted the following experiments and shown the simulation results as follows.

We first measure how the number of established groups of each algorithm varies with the total number of MANET members, which is represented as the index of the average number of established groups. In another set of experiments, we would conduct the simulation where we estimate the average group establishing time for each algorithm under different network densities, which can be assessed as the sensitivity of the grouping algorithm.

With regard to node mobility, we conduct experiments assuming the members are moving according to the random waypoint (RW) model with random pause time in a region of $1{,}000 \times 1{,}000$ m. The mobile speed for each member is set to 5 to 20 m/s. The transmission range for members is set to 100 and 200 m, respectively, for comparison. To perform each algorithm in a strategic mobile scenario, we set MANET members' mobile probability being proportional to their density (i.e., $\Pr_i^{\text{mob}} \propto \text{Den}(i)$) computed by the equation $\Pr_i^{\text{mob}} = {}^{\text{Den}}(i)/_n$, where n denotes the number of MANET members. At each simulation during the same parameter settings, we performed 500 rounds and recorded each involved index. The final values in the following figures are the mean value of these indexes.

From Figures 6 and 7, we can see that the increase in transmission range of MANET members serves the purpose to reduce the average number of established groups. Take our method, for example, in a situation where the network density stays at a low level (i.e., the number of MANET members is less than 200), when the transmission

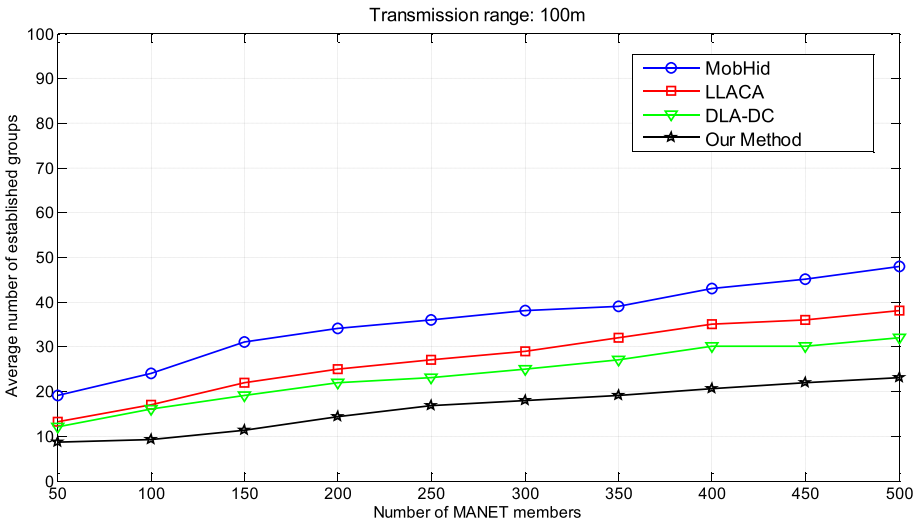

Figure 6 Comparison of average number of established groups (transmission range: 100 m).

range changes from 100 to 200 m, the average number of established groups has gone down from 10.85 to 4.2 with a 61.3% drop. Similarly, in a situation where the network density stands at a relatively high level (i.e., the number of MANET members ranges from 200 to 500), the same index has gone down from 19.88 to 6.42 with a 67.7% drop. We can also see that the average number of established groups of each algorithm increases with different degrees as the density of MANETs increases. For comparison, as seen from Figure 6 showing the situation where the transmission range is 100 m, when the number of MANET members ranges from 50 to 500, the established group number of MobHid increases on average from 19 to 48, while the same index of LLACA and DLA-DC increases from 13.1 and 12 to 38 and 32, respectively. On equal terms, the average number of established groups of our method only increases from 8.6 to 23; in addition, when the member number is less than 300, this growth is very slow. All these trends exist in a situation where the transmission range is 200 m as represented in Figure 7. Compared with the other three schemes, our method has significantly reduced the average number of established groups so as to reduce the group maintenance and management overhead.

Figures 8 and 9 show the average group establishing time of each algorithm under the two situations where the members' transmission range is 100 and 200 m, respectively. From these figures, we can find that adopting different algorithms the average group establishing time raises as the number of MANET members increases, and its growth from large to small order is MobHid, DLA-DC, LLACA, and our method. Because of maintaining and

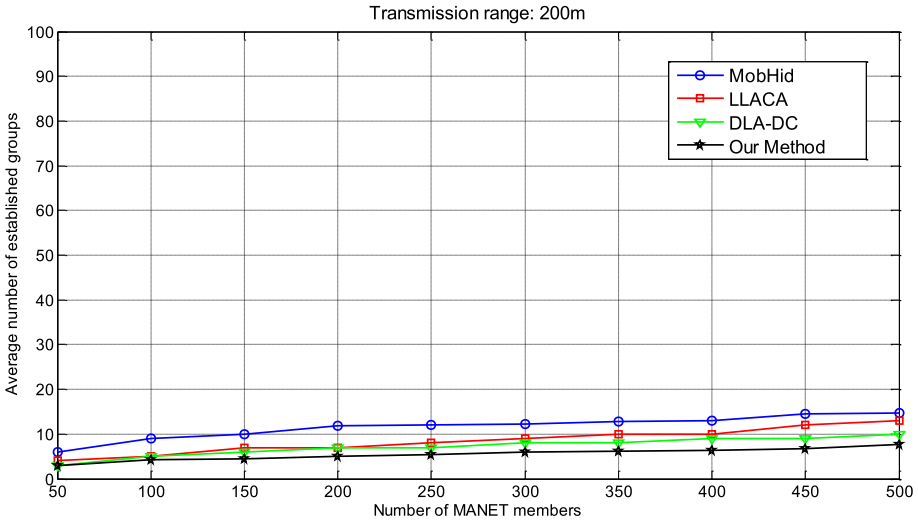

Figure 7 Comparison of average number of established groups (transmission range: 200 m).

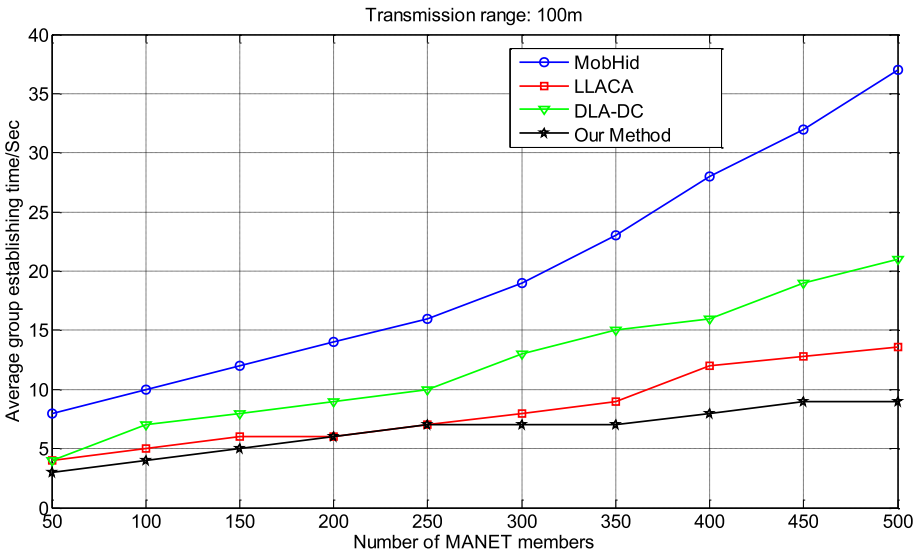

Figure 8 Comparison of average group establishing time (transmission range: 200 m).

interacting only with the neighbors, our method performs best of the four involved algorithms for the average group establishing time. As we know, the average group establishing time can be evaluated as grouping efficiency and algorithm sensitivity. In MANETs, the requirement of self-organized information processing for mobile members can be satisfied if the grouping procedure is finished as quickly as possible; what is more, a sensitive grouping algorithm can contribute to reduce the redundant information generated by the group re-establishment procedure caused by node mobility. So compared with the other three schemes, our method plays the most significant role to enhance the sensitivity and efficiency of the grouping algorithm in the strategic mobile scenario of MANETs.

5.2.2 Performance analyses for group key management in the strategic mobile scenario of MANETs

Aiming at the group key management in strategic mobile scenario of MANETs, our method can efficiently realize the anonymous communication tolerating part of members moving across multigroups. The mechanism of our method is to build the mobile management model to monitor the roaming nodes within the adjacency field for the purpose of reducing the key updating frequency and establishing latency.

Finally, we have designed several experiments to analyze the performance of our key management protocol. In addition, we implement the other two representative group key management protocols of MANETs for

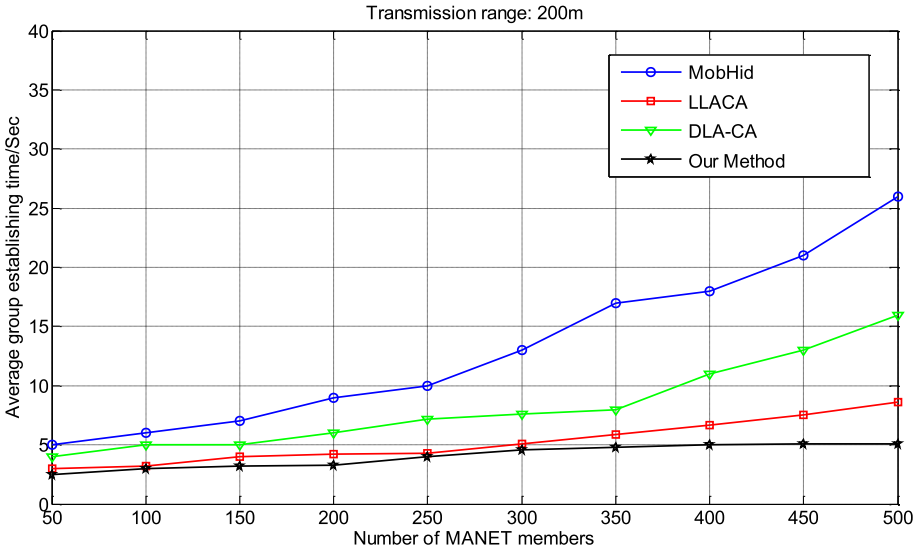

Figure 9 Comparison of average group establishing time (transmission range: 200 m).

comparison, which are Zhu's threshold hierarchy scheme [15] and JHc's GDH3-based region key management scheme [16,17], respectively. Similarly, we have employed the RW model with the mobile probability following the equation $\Pr_i^{\mathrm{mob}} = {}^{\mathrm{Den}}(i)/_n$ to simulate member's mobility in strategic scenario. In this series of experiments, to measure the impact of the different mobile speeds and communication environments on each protocol, we run the experiments under three mobile levels (5 to 10, 10 to 15, and 15 to 20 m/s, respectively) and four communication environments represented as link failure rate (LFR; 5%, 10%, 15%, and 20%, respectively). The other parameters have set as follows: We set the number of MANET members changing from 50 to 500. The transmission range for each member is 100 m; the key size is 1,024 bits. At each simulation during the same parameter settings in Section 4.2.2 we performed it 500 rounds and recorded the following indexes. The final values in each figure are the mean value of these indexes. Note that each round was finished when 70% of MANET members have exhausted their energy.

1. Average number of updates for group session key (AUK)
2. Average number of group operations (ANO)
3. Average network lifetime (ANL)

For comparing the index AUK of each protocol exactly, when terminating the grouping operation, we set the core region (i.e., the group manager together with its one-hop neighbors) of each group remaining stationary just only to evaluate the update number of the group session key by using the above different protocols.

In Zhu's scheme, mobile nodes have been divided into several hierarchy trees, similar to the groups of our protocol, according to the pre-set value of the regional trust coefficient (RTC) and global trust coefficient (GTC), where the leaf parts of the tree are those real nodes and the rest parts denote the virtual nodes responding to manage and distribute the shared key. For a real node with sufficient storages, it will store two kinds of public keys: (1) public keys of itself and virtual nodes that are on its reverse path to the root, which can be denoted by the group of all these nodes, and (2) public keys of nodes within the same region of any member of the group. Hence, according to this scheme, for guaranteeing a highly secure environment (i.e., the probability of node compromising and group compromising is less than 5% and 0.01%, respectively), we set different secret thresholds represented as k towards different numbers of MANET members in our experiment listed in Table 2.

Table 2 The setting of secret thresholds used in Zhu's scheme

Number of MANET members	Value of k
50	13
100	16
150	20
200	25
250	28
300	31
350	35
400	39
450	42
500	45

In JHc's scheme, due to lack of a self-organized grouping mechanism we adopted the optimum region sets as the group establishing parameters for a comparable experiment of this part. More specifically, the field of 1,000 m × 1,000 m was zoned equally as 37 hexagon groups geographically which can result in an efficient trading inter-regional vs. intra-regional group key management overheads.

We have conducted simulation experiments of the above algorithms. All experiment results are computed on a DELL OptiPlex 360 Desktop PC with Intel Core™ 2 Duo 2.66-GHz E7300 processor and 2,048-MB RAM. (Dell Inc., Round Rock, TX, USA). The series of experiments to evaluate the AUK for each algorithm within the network lifetime have been conducted, and the results are shown in Figure 10. We can find that in the strategic mobile scenario of MANETs, the member's mobile speed, the LFR and the network density would affect the value of AUK. The first observation is that the value of AUK of each algorithm rises as the LFR increases. Because when the LFR increases, the probability of members failing to communicate with each other, the normal node being mistakenly detected to the malicious node, and the network energy being quickly exhausted caused by multiple message retransmission requests inevitably turns to be larger. For clarity, take Zhu's scheme with the third level of member mobile speed (i.e., ML3) for example, in a situation where the number of MANET members is 300, when the network LFR stays at 5%, 10%, 15%, and 20%, respectively, the value of AUK is 48.6, 50, 71.9, and 78.4. The second observation is that when fixing the LFR, the value of AUK of each algorithm turns to decline, to some extent, as the network density increases. The reason is that a high network density stands for a large probability of network tending to be stable. Hence, in MANETs with a high density, the updates led by topology changes are no longer in a dominant position compared to those by periodical updates. Under such circumstances the value of AUK of each algorithm tends to be the product between the update

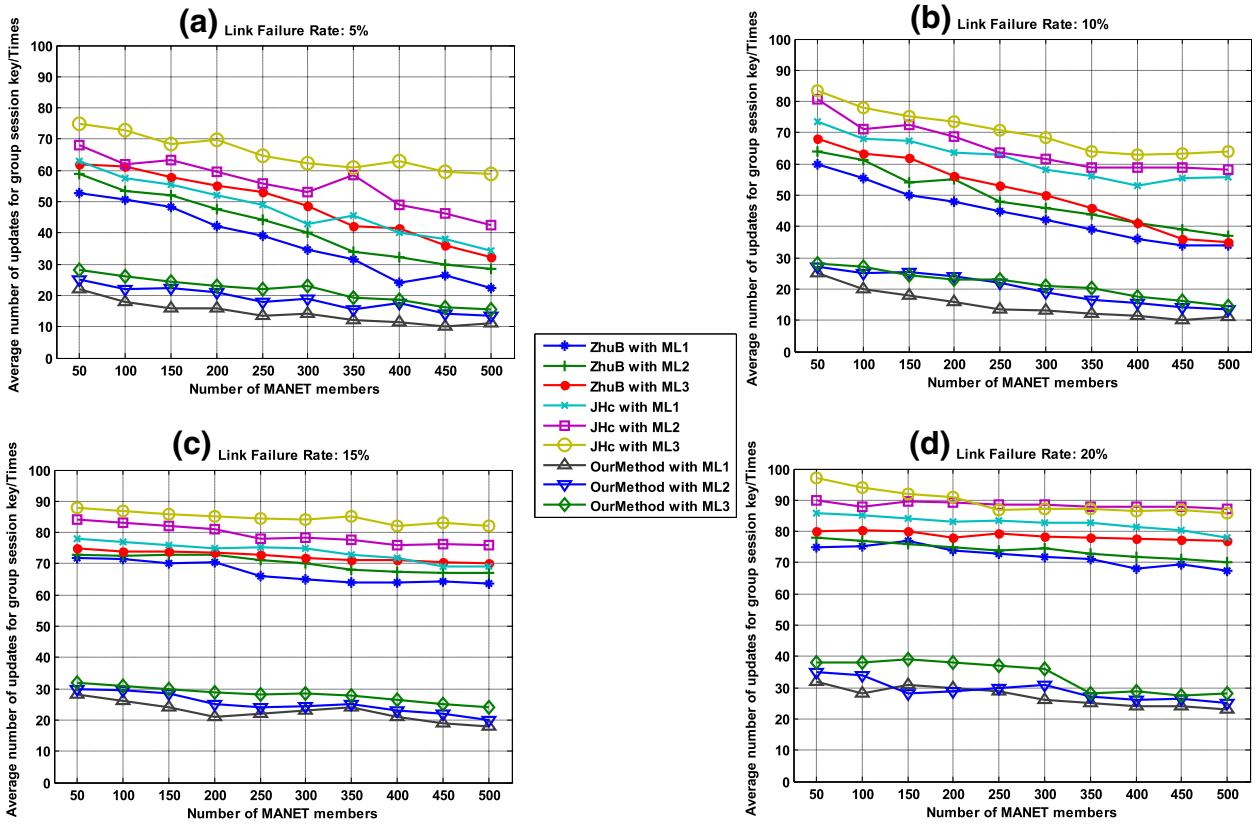

Figure 10 Comparison of AUK. Link failure rates: **(a)** 5%, **(b)** 10%, **(c)** 15%, **(d)** 20%.

period and the network lifetime, represented as AUK = $T_{\text{update}} \times \text{ALF}$. The third observation is that when the network member number and the LFR are under the same condition, the value of AUK of each algorithm turns to increase as ML upgrades. For example, when member number is 400 and the LFR is 20%, in a situation where the mobile level of members in the network are set to be ML1, ML2, and ML3, the value of AUK is 24, 26, and 29, respectively. To sum up, compared with the other two algorithms, our method can effectively reduce the value of AUK (i.e., reduce the key updating overhead) and tolerate part of failure of communication as well as high speed mobility of network members, which results in an improvement of the real-time and secure indexes.

Figure 11 shows the result of the ANO for each algorithm within the network lifetime under various experiment conditions. In these experiments, the group operation consists of three behaviors including group merging, group partition, and re-grouping. More specifically, in Zhu's scheme, all three behaviors were defined and handled by designed procedures. In JHc's scheme, group merging and group partition are the main forms of the group operation, while in our method, the re-grouping operation is the only involved way to the group operation. Because the group operation is the most resource-intensive behavior for group key management protocol, measuring the value of ANO of each algorithm can exactly assess their communication, computation, and storage overhead. From Figure 11, we can find that first, under the same LFR, the value of ANO of each algorithm rises with a slight fluctuation as the network density increases. Second, under the same situation of network density, the value of ANO of each algorithm rises as the mobile level of nodes upgrades from ML1 to ML3. Third, under the same mobile level together with the network situation, the value of ANO of each algorithm rises as the LFR ranges from 5% to 20%. To sum up, compared with the other two schemes, our method could reduce the value of ANO always being the lower level, which reflected that our group key management protocol can effectively save the communication, computation, and storage overhead in the strategic mobile scenario of MANETs.

Finally, we ran the experiments to measure the ANL of each algorithm under different conditions of network density and LFR. Note that in these series of experiments, the mobile speed of MANET members is set to be 5 to 20 m/s. From Figure 12, we can see that as the LFR increases, under the same condition of network density, the

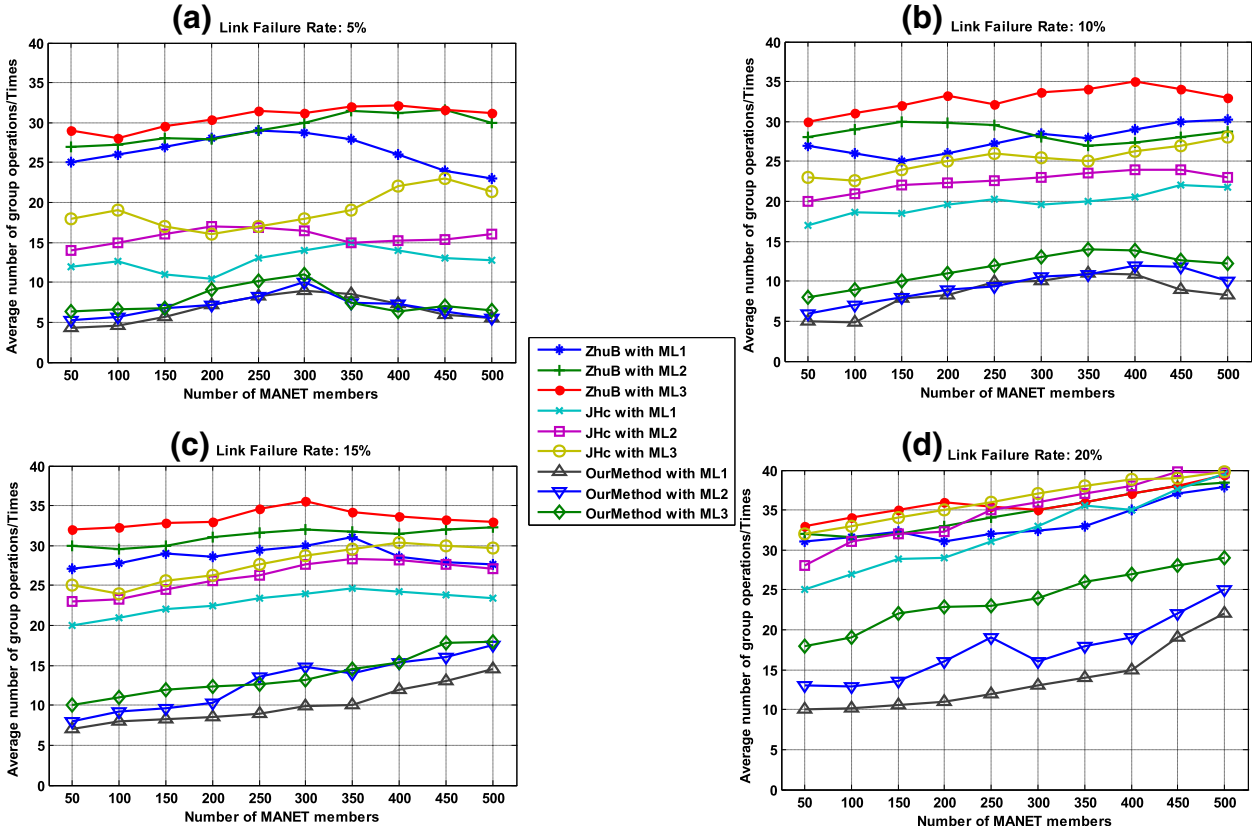

Figure 11 Comparison of ANO. Link failure rates: **(a)** 5%, **(b)** 10%, **(c)** 15%, **(d)** 20%.

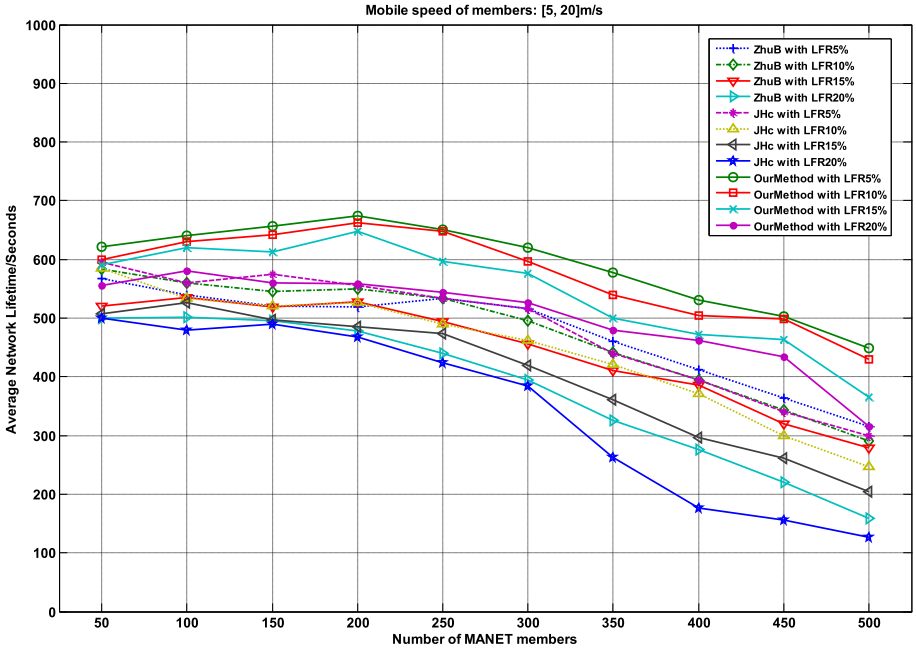

Figure 12 Comparison of ANL.

value of ANL of each algorithm appears to decline with different degrees. The reason is that due to a higher LFR, the probability of message retransmission turns to be larger which can result in a rapid exhaustion of members' energy. Take JHc's scheme for example, when the number of MANET members is 400, in a situation where the LFR is 5%, 10%, 15%, and 20%, respectively, the value of ANL is 394, 371, 297, and 176 s, which presents a decline tendency. In addition, as the scale of the network enlarges (i.e., the network density increases) under the same conditions, the value of ANL of each algorithm declines with different degrees. For example, when LFR is 10%, the value of ANL in Zhu's scheme has gone down from 584 to 290 with a 50.3% drop. The same index in JHc's scheme has gone down from 585 to 246 with a 57.9% drop, while that in our method has gone down from 600 to 430 with a mere 28.3% drop. Hence, compared with the other two schemes, not only in terms of absolute value but also the relative value for the decline, our method shows the best performance to prolong the network lifetime. What is more, the experiment results can also demonstrate that when the network member number increases from 50 to 500, the lowest drop of ANL in our method can satisfy the requirement of MANET scalability which plays an important role to prolong the network lifetime.

6 Conclusions

In this paper, we have proposed an energy-efficient group key management protocol for the strategic mobile scenario of MANETs. It mainly depends on the three proposed mechanisms to address the problem of enhancing energy-efficient security and scalability performance for the protocol handling key establishment and distribution in the strategic mobile scenario of MANETs. Both theoretical analyses and simulation experiments have demonstrated that our protocol is more scalable to the strategy mobile application scenario of MANETs with a large number of users and enables energy-efficient security without increasing computation and communication overhead. In contrast, we have distinguished typical grouping methods from a performance point of view and shown that our group establishing algorithm is the best to satisfy the requirement of computational complexity and convergence for MANETs. We also have compared our key management protocol with other three schemes and indicated that our method performed best that can meet the scalability requirement which plays an important role in prolonging the network lifetime and effectively save the communication, computation, and storage overhead in the strategic mobile scenario of MANETs.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work is supported by the Science and Technology Foundation of Guizhou Province (Grant No. QianKeHe J [2013] 2117), the Industrial Research project of Guizhou Province (Grant No. QianKeHe J [2013] 3061), and the Research Fund for the Doctoral Program of Guizhou University (Grant No. GuiDaRenJiHeZi [2011] 19).

Received: 15 April 2014 Accepted: 28 August 2014
Published: 6 October 2014

References

1. M Omar, Y Challal, A Bouabdallah, Certification-based trust models in mobile ad hoc networks: a survey and taxonomy. J. Netw. Comput. Appl. **35**(1), 268–286 (2012)
2. MA Moharrum, AA Al-Daraiseh, Toward secure vehicular ad-hoc networks: a survey. IETE Tech. Rev. **29**(1), 80–89 (2012)
3. JH Cho, IR Chen, PG Feng, Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad-hoc networks. IEEE Trans. Reliab. **59**(1), 231–241 (2010)
4. D Subhadrabandhu, S Sarkar, F Anjum, Efficacy of misuse detection in ad-hoc networks, in *1st Annual IEEE Communications Society Conference on Sensor and Ad-hoc Communications and Networks* (Santa Clara, 2004), pp. 97–107
5. MY Su, Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. Comput. Commun. **34**(1), 107–117 (2011)
6. F Renjian, X Xiaofeng, Z Xiang, A trust evaluation algorithm based on node behaviors and D-S evidence theory for wireless sensor networks. Sensors **11**(2), 1345–1360 (2011)
7. A Rehan, K Turgay, GV Raju, EMLTrust: an enhanced machine learning based reputation system for MANETs. Ad Hoc Netw. **10**(3), 435–457 (2012)
8. PB Velloso, RP Laufer, D de o Cunha, OCMB Duarte, G Pujolle, Trust management in mobile ad hoc networks using a scalable maturity-based model. IEEE Trans. Netw. Serv. Manag. **7**(3), 172–185 (2010)
9. D Chen, GR Chang, DW Sun, J Li, J Jia, X Wang, Comput. Sci. Inform. Syst. **8**(4), S1, 1207–2228 (2011)
10. G Saurabh, KB Laura, BS Mani, Reputation-based framework for high integrity sensor networks. ACM Trans. Sensor Networks **4**(3), 1–37 (2008)
11. TH Lacey, RF Mills, BE Mullins, RA Raines, ME Oxley, SK Rogers, RIPsec - using reputation-based multilayer security to protect MANETs. Comput. Security **31**(1), 122–136 (2012)
12. G Tuna, SM Potirakis, G Koulouras, Implementing a trust and reputation model for robotic sensor networks. Elektronika Ir Elektrotechinka **19**(10), 3–8 (2013). ISSN 1392–1215. doi:10.5755/j01.eee.19.10.5884
13. MMEA Mahmoud, XM Shen, FESCIM: fair, efficient, and secure cooperation incentive mechanism for multihop cellular networks. IEEE Trans. Mob. Comput. **11**(5), 753–766 (2012)
14. K Dirra, H Seba, H Kheddouci, ECGK: an efficient clustering scheme for group key management in MANETs. Comput Commun **33**, 1094–1107 (2010)
15. B Zhu, F Bao, RH Deng, MS Kankanhalli, G Wang, Efficient and robust key management for large mobile ad hoc networks. Comput. Netw. **48**, 657–682 (2005)
16. JH Cho, IR Chen, DC Wang, Performance optimization of region-based group key management in mobile ad hoc networks. Perform. Eval. **65**, 319–344 (2008)
17. JH Cho, IR Chen, Performance analysis of hierarchical group key management integrated with adaptive intrusion detection in mobile ad hoc networks. Perform. Eval. **68**, 58–75 (2011)
18. DJ Huang, D Medhi, A secure group key management scheme for hierarchical mobile ad hoc networks. Ad Hoc Netw. **6**, 560–577 (2008)
19. I Khalil, S Bagchi, C Rotaru, N Shroff, UnMask: utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks. Ad Hoc Netw. **8**(2), 148–164 (2010)
20. I Khalil, S Bagchi, Stealthy attacks in wireless ad hoc networks: detection and countermeasure. IEEE Trans. Mob. Comput. **10**(8), 1096–1112 (2011)
21. F Adelantado, C Verikoukis, A non-parametric statistical approach for malicious users detection in cognitive wireless ad-hoc networks, in *Proceedings of 2011 IEEE International Conference on Communications* (ICC, Kyoto, 2011)
22. C-T Li, C-C Yang, M-S Hwang, A secure routing protocol with node selfishness resistance in MANETs. Int. J. Mobile Commun. (IJMC) **10**(1), 103–118 (2012)

23. C Konstantopoulos, D Gavalas, G Pantziou, Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction. Comput. Netw. **52**, 1797–1824 (2008)
24. JA Torkestani, MR Meybodi, Clustering the wireless ad hoc networks: a distributed learning automata approach. J. Parallel Distrib. Compt. **70**, 394–405 (2010)
25. JA Torkestani, MR Meybodi, LLACA: an adaptive localized clustering algorithm for wireless ad hoc networks. Comput. Electr. Eng. **37**, 461–474 (2011)
26. B Mohamed Salah, C Isabelle, F Olivier, Group key management in MANETs. Int. J. Network Security **6**(1), 67–79 (2008)
27. A Abdel-Hafez, A Miri, L Orozco-Barbosa, Authenticated group key agreement protocols for ad hoc wireless networks. Int. J. Netw. Secur. **4**(1), 90–98 (2007)
28. X Wang, YF Wu, N Yu, JW Wan, A self-adaption link-quality detection algorithm for data collecting in OSN, in *Proceedings of 2010 IEEE Asia-Pacific Services Computing Conference (APSCC 2010)* (Hangzhou, 2010), pp. 516–522
29. XL Huang, FY Ma, WJ Zhang, TAON: a topology-oriented active overlay network protocol, in *6th International Conference on Mobile Adhoc and Sensor Systems* (Macau, 2009), pp. 884–890
30. SM George, W Zhou, H Chenji, M Won, YO Lee, A Pazarloglou, R Stoleru, P Barooah, DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response. IEEE Commun. Mag. **48**(3), 128–136 (2010)
31. WF Liang, B Richard, YL Xu, Q Wang, Minimum-energy all-to-all multicasting in wireless ad hoc networks. IEEE Trans. Wirel. Commun. **8**(11), 5490–5499 (2009)
32. S Guo, VCM Leung, A distributed algorithm for min-max tree and max-min cut problems in communication networks. IEEE/ACM Trans. Networking **18**(4), 1067–1076 (2010)

doi:10.1186/1687-1499-2014-161
Cite this article as: Wang *et al.*: **The energy-efficient group key management protocol for strategic mobile scenario of MANETs.** *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:161.