Incentive-Based Optimal Nodes Selection Mechanism for Threshold Key Management in MANETs with Selfish Nodes

Yuanbo Guo,1,2 Jianfeng Ma,3 Chao Wang,3 and Kuiwu Yang1

1 State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450004, China
2 Science and Technology on Information Assurance Laboratory, Beijing 100072, China
3 Computer Science School of Xidian University, Xi'an 710071, China

Correspondence should be addressed to Yuanbo Guo; yuanbo_g@hotmail.com

Received 28 February 2013; Revised 30 March 2013; Accepted 8 April 2013

Academic Editor: Liangmin Wang

Copyright © 2013 Yuanbo Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most of the previous work on threshold-cryptography-based distributed CA concentrates on the initial systems configurations and concrete protocols design, ignoring the efficiency and effectiveness of the key management service during its operation, and always assuming that there are honest nodes to carry out the service faithfully. This paper focuses on developing a selection mechanism in MANETs with selfish nodes, to dynamically select a coalition of nodes carrying out the threshold key management service optimally during system operation. First, we formulate the dynamic nodes selection problem as a combinatorial optimization problem, with the objectives of maximizing the success ratio of key management service and minimizing the nodes’ cost of security and energy. Then, to ensure truth telling is the dominant strategy for any node in our scenario, we extend the payment structure of the classical Vickrey-Clarke-Groves (VCG) mechanism design framework and divide the payment into pieces to the nodes, with the consideration of their actual execution effectiveness. Simulations show that the proposed mechanism enjoys improvements of both the success ratio of key management service and lifetime of the network, as well as reductions of both the cost of participating nodes and compromising probability of MANETs, compared with the existing work.

1. Introduction

A mobile ad hoc network (MANET) is a network consisting of a collection of nodes capable of communicating without relying on a fixed infrastructure and is characterized by some of the features like lacking infrastructure, dynamic network topology, distributed operation, variable capacity links, use of low power devices, and so forth. This makes ad hoc networks financially viable and have tremendous potential for communications in battlefields, disaster recovery areas, and other environments such as collaborative computing and communications in smaller areas. For MANETs, public-key cryptography (PKC) is appealing in offering security support, due to its effectiveness in facilitating essential security services such as digital signatures and key management. However, the traditional public key infrastructure (PKI) supporting key management approaches require a global trusted certificate authority (CA) to manage public key certificates used to generate confidence in the legitimacy of public keys for the nodes of the network. This makes it difficult to deploy the PKI in MANETs, since this type of networks does not have any form of online or offline authority [1]. Even if the service node can be defined to act as an authority, maintaining such a centralized server and keeping its security and availability in such a dynamic network is a difficult task. Key management for MANETs therefore needs to mitigate the unreliability of basic CA services by taking on a distributed, self-organizing nature [2–9].

However, previous work on this subject mainly concentrates on the initial systems configuration and concrete protocols design of distributed CA itself and ignores the problem of how to select a threshold number of nodes from the set of all partial certificates during its operation with the consideration of attributes of all nodes in the network. Instead, a random selection scheme is often assumed or implicated. To the best of our knowledge, the only paper addressing the problem of optimal nodes selection for threshold key management in MANETs is [10], where the dynamic nodes selection process is formulated as a multiarm bandit problem. Then, an optimal selection scheme is proposed to select the best nodes to be
used as private key generators (PKGs) from all available ones with the consideration of their security conditions and energy states. This scheme has nice features of decreasing network compromising probability and increasing network lifetime in MANETs. There are still some problems suffering from the existing schemes, including the one proposed in [10]: (i) they do not consider the effectiveness of the key management. Given a crypto threshold $k$, more than $k$ correct replies from nodes make a key management service successful. The success ratio must be kept at a high level under all circumstances to provide useful and effective key management services; (ii) they always assume that the nodes in MANETs cannot act rationally and strategically (i.e., each node follows the protocol specification by assumption).

In this paper, we present incentive compatible optimal nodes selection (ICONS), a mechanism which dynamically implements the optimal nodes selection for threshold key management based on the nodes’ security and energy states truthfully in dominant strategies. Specifically, we formulate the dynamic nodes selection problem as combinatorial optimization problem [11], by combining two objectives of maximizing the success ratio of key management service and minimizing the nodes’ cost of security and energy into a single weighted objective firstly. And then we extend the classical Vickrey-Clarke-Groves- (VCG-) [12] based mechanism design framework [13] to allow for implementing an objective function which is not quasi-linear and then divide the payment into pieces to the nodes according to their outcomes at current stage. The proposed mechanism not only enjoys the same nice features as the scheme in [10] (decreasing network compromising probability, lowering the energy cost, and prolonging network lifetime) but also achieves more performance benefits of increasing the success ratio of key management service and allowing the nodes in MANETs to remain truthful in the scenario where they act rationally and selfishly.

The rest of the paper is organized as follows. The next section reviews related work. Section 3 formulates the optimal nodes selection model for threshold key management in MANETs. Section 4 presents the incentive compatible optimal nodes selection mechanism and proves its correctness and truthfulness. The performance of our model is evaluated via detailed simulations in Section 5. Finally, this paper is concluded and discussed in Section 6.

2. Related Work

In this section, we review the related work in threshold-cryptography-based distributed CA (DCA) and mechanism design application in MANETs.

2.1. Threshold-Cryptography-Based DCA in MANETs. A DCA is realized through the distribution of the CA’s private key to a number of shareholding nodes. The design of a DCA based on threshold cryptography is suggested in [14] firstly and then applied to solving the key management problem in MANETs in [2] by letting a set of nodes in the network share the system secret. From then, many DCA schemes in MANET have been proposed, which can be classified as partially or fully distributed certificate authorities [15].

In partially implemented DCA, services of the CA are distributed to a set of specialized server nodes using secret sharing. Each of these nodes can generate partial certificates and a user can create a valid certificate by combining enough number of these partial certificates. In [5], a cluster-based partially DCA architecture in MANETs is established. First, a cluster head assisted CA locating scheme is proposed to shift the responsibility of CA discovery from each user node to cluster heads, which greatly reduces service response time and system overhead. Then, a share update procedure is also proposed to resolve the multiple initializations problem and achieves fast systemwide update. The authors of [6] propose a partially distributed certificate management mechanism that can handle mobility of nodes for MANET. The mechanism segregates the roles of certification authority to keep with the dynamic mobility of nodes and handle rapid and random topological changes with minimal overhead. The mobile certificate authority (MOCA) key management framework is proposed in [7] based on threshold cryptography to provide authentication service for MANETs. MOCA utilizes a carefully selected set of mobile nodes to function as a collective certificate authority while the MOCA nodes are kept anonymous. Equipped with a novel routing protocol designed to support the unique communication pattern for certification traffic, MOCA achieves high availability key management and authentication service with intuitive metrics to measure the provided quality of service. Then, the authors of [16] extend the MOCA framework by proposing and evaluating a key management scheme that suits the dynamic nature of an ad hoc network. To enhance the robustness and security of the threshold key management scheme, the authors of [4] propose a secure and robust key management scheme (SRKM) based on threshold cryptography, making it more difficult for mobile adversaries to violate the secrecy of the private key of certification service, even if they compromise more than a threshold number of nodes.

In fully distributed CA, services of a CA are distributed to all nodes using secret sharing, and each of these nodes can generate partial certificates. Since almost all the neighbors of a requesting node hold shares of the DCAs private signature key, fully distributed CA reduces the communication delay and improves the availability. The authors of [8] distribute the functionality of conventional security servers, specifically the authentication services, so that each individual node can potentially provide certification services for other nodes in MANETs. Centralized management is minimized and the nodes in the network collaboratively self-secure themselves. Then, the authors of [17] propose a modification to the scheme in [8] to make it suitable for a mobile ad hoc network in which forming a coalition of a large number of nodes is often difficult. The concept of redundancy in key shares is introduced to increase the probability of recreating the CA key for a node in a highly mobile network, by allocating more than one share to each node. In [3], a scheme called autonomous key management (AKM) is proposed to provide a self-organizing and fully distributed key management service, which uses hierarchical structure to ensure flexibility and adaptability.
and uses verifiable secret sharing (VSS) to resist active attacks. The authors of [9] propose a fully distributed trust model based on trust graph for mobile ad hoc networks, where nodes have a similar role and do not need to assign any special functions to a subset of nodes. This scheme allows users to fully control the security settings in the network and allows nodes to generate, store, and distribute their public key certificates without any central server or trusted party. The scheme is developed for open networks, in which nodes can join/leave the network without any centralized administration. The joining operation is performed by a coalition of member nodes to allow access to a new node.

2.2. Application of Mechanism Design. Mechanism design is the subfield of microeconomics and game theory that considers how to implement an optimal systemwide solution to problems that involve multiple self-interested players, each with private information about their preferences for outcomes [13]. It is a useful and powerful tool to design protocols in the environment where the players may deviate the given protocol specification if it is beneficial for them to do so, and has been used extensively in MANETs environments.

The work in [18] proposes ad hoc VCG, a reactive routing protocol for MANETs that is robust against individual selfishness of the communication nodes and achieves cost-efficiency and truthfulness. This scheme works well in the MANETs environment, where the communication nodes are assumed to be selfish and need to declare their cost of energy in order to compute a cost-efficient communication path. Following this approach, the authors of [19] present low overhead truthful routing protocol (LOTTO), a low-overhead truthful routing protocol for route discovery in MANETs with selfish nodes by designing incentives based on VCG mechanism [12], to prevent nodes from revealing fake information and ensure truth telling to be the dominant strategy among all nodes. In [20], a mechanism-design-based model is proposed to motivate nodes that do not belong to the confident community to participate in being selected as RA, by giving them incentives in the form of trust. An RA selection algorithm is also proposed in this paper to select nodes based on a predefined selection criteria function and nodes location. In [21], a novel surveillance mechanism is proposed to observe the packet-dropping behavior of suspicious insiders. It quantifies the threat level of the suspicious insiders and then realizes an incentive-compatible surveillance scheme to motivate the rational monitors to cooperate, by rewarding the cooperating monitors and punishing the violating monitors.

The authors of [22] study the leader election in the presence of selfish nodes for intrusion detection in MANETs and propose an integrated solution for prolonging the lifetimes of mobile nodes and prevent the emergence of selfish nodes. Reputations are computed in [22] also by using the well-known VCG [12] mechanism design as a theoretical tool.

These existing studies clearly show that mechanism design becomes prevalent in many engineering applications in MANETs. It provides a rich set of mathematical tools and models to motivate the nodes to reveal truthfully their selection criteria function. However, there is no much work on applying mechanism design theory to threshold key management in MANETs, where the success of the key management task is highly dependent upon the distributed collaboration of a coalition of rational and selfish nodes.

3. Optimal Nodes Selection Model

3.1. System Models. MANET, in this study, is represented by an undirected graph \( \mathcal{G} = (\mathcal{N}, \mathcal{L}) \), where \( \mathcal{N} = \{1, 2, \ldots, N\} \) is the set of wireless nodes including a leader, and \( \mathcal{L} \) is the set of communication links between the nodes. To keep the security and energy information current, we divide the time axis into stages, which correspond to the time intervals between two continuous key management tasks of the MANET. The stages are indexed by the integers, so in each stage \( t \) only one threshold key management task can be completed by a coalition of nodes cooperatively.

At each stage, the leader takes the role of selecting the best nodes from \( \mathcal{N} \) to act as server nodes based on the security and energy states of each node, and pays each selected node according to its completion of assigned task or not. Each server node has its own share of the private CA key and participates in the process of threshold key management in the current stage. Sometimes, we also call a server node a active node and call a nonserver node a passive node. The leader is not necessarily a special node in MANET. Instead, it can be elected dynamically from the nodes set \( \mathcal{N} \) by a leader election algorithm. The aim of a leader election algorithm is to ensure that a suitable node in a network will be selected as the leader to perform a task whenever needed [23]. Since in this paper we mainly focus on developing an incentive compatible optimal nodes selection mechanism, to encourage each node in the system to be truth-telling, the details of the leader election are out of the scope here. There are several leader election researches that have been done for MANETs and wireless sensor networks [22, 24, 25].

3.1.1. Security Model. Denote by \( s_i^t \) the security state of a node \( i (i \in \mathcal{N}) \) at the stage \( t \). Security state represents the security condition of a node and can be monitored by node itself with a local intrusion detection system (IDS). Assume each node \( i \) has a finite number of \( \mathcal{I}_i \) states in security state space \( \mathcal{S} \), and each security state \( s_i^t \) evolves according to an \( \mathcal{I}_i \)-state Markov chain with one-step transition probability matrix as follows:

\[
A_i^a = (\phi_{jk})_{j,k \epsilon \mathcal{I}_i} = \Pr (s_i^{t+1} = k | s_i^t = j),
\]

where \( a \in \{0, 1\} \) stands for an action. Action 1 means that the node is active, and action 0 means that the node is passive.

3.1.2. Energy Model. We represent each node \( i \)'s energy state at stage \( t \) as \( e_i^t \), which can also be detected locally. We assume that the continuous battery residual energy can be divided into discrete levels, denoted by \( \mathcal{E} = (e_1, e_2, \ldots, e_{\ell}) \). To foresee the energy consumption at the current stage, we model the transition of energy levels of nodes in MANETs as a Markov chain with one-step transition probability matrix as follows:

\[
B_i^a = (\psi_{jk})_{j,k \epsilon \mathcal{E}} = \Pr (e_i^{t+1} = k | e_i^t = j).
\]
3.1.3. Node States Model. Note that both security state $s_i$ and energy state $e_i$ are independent of each other, so we can model the state of each node $i$ at stage $t$ as

$$\pi_i^t = (s_i^t, e_i^t).$$

The state set of $\pi_i^t$ is represented as $\Pi_i$ and we have $|\Pi_i| = |\mathcal{S}| \times |\mathcal{E}|$. The state $\pi_i^t$ evolves with one-step transition probability matrix as follows:

$$P_i^t = [A_i^t \otimes B_i^t],$$

where $A_i^t$ is security state transition probability matrix, $B_i^t$ is energy states transition probability matrix, and $\otimes$ denotes the Kronecker product.

3.1.4. Cost Model. The costs associated with each node $i$ at stage $t$ are defined as security cost $\mu_i(s_i^t, a_i^t)$, from a potential compromise of the node, and the energy cost $\mu_e(e_i^t, a_i^t)$. The action adopted by node $i$ at stage $t$ is denoted as $a_i^t \in \{0, 1\}$.

Then, the instantaneous cost of node $i$ at stage $t$ is

$$z_i^t(s_i^t, e_i^t, a_i^t) = (1 - \gamma) \cdot \mu_i(s_i^t, a_i^t) + \gamma \cdot \mu_e(e_i^t, a_i^t),$$

where $\gamma \in (0, 1)$ is the weight factor for the two kinds of costs and could be adjusted according to circumstances. We then extend the cost model to more realistic settings by considering the network lifetime as follows:

$$z_i^t(s_i^t, e_i^t, a_i^t) = \frac{(1 - \gamma) \cdot \mu_i(s_i^t, a_i^t) + \gamma \cdot \mu_e(e_i^t, a_i^t)}{e_i^t},$$

where $e_i^t$ is node $i$'s energy level at stage $t$.

Since just one message is needed for a passive node to report its states to the leader, the cost of a passive node can be assumed as a constant in a given stage $t$. Then, the node $i$'s cost at stage $t$ can be denoted as

$$c_i^t(s_i^t, e_i^t) = \begin{cases} z_i^t(s_i^t, e_i^t, 1) & \text{if } i \text{ is a active node} \\ C^t & \text{if } i \text{ is a passive node}, \end{cases}$$

where constant $C^t$ is the cost of a messages transmission in stage $t$.

If there are $m$ active nodes at stage $t$ (when $(m, N)$ secret sharing is used in the threshold key management scheme), then the cost of all the nodes for key management is

$$Z(t) = \sum_{i=1}^{N} c_i^t(s_i^t, e_i^t, 1) = \sum_{j=1}^{m} z_j^t(s_i^t, e_i^t, 1) + (N - m) \cdot C^t,$$

where $\{i_1, i_2, \ldots, i_m\} \subset \mathcal{N}$ denotes the set of all active nodes at stage $t$.

3.1.5. System Value Model. Key management service consists of a set of tasks and procedures supporting the establishment and maintenance of keying relationships between authorized parties [27], such as new node authentication and admission, generation and distribution keying material, update/revocation/destruction of keying material, bootstrapping, and maintenance of trust in keying material. Without loss of generality, we assume that the key management task at each stage $t$ has a certain value $V^t$ to the system, which is determined by the expected gain that system could gain from successful completion of this task. For example, the task of joint authentication has a higher gain for the system to admit a new node to join the network than that of joint session key establishment for two existing peer nodes, because joint authentication might enlarge the network scale. So, the former can be assigned a value of 200 while the latter just be assigned 120.

In the presence of attacks, an active node may fail to complete its assigned task of key management to act as a server node. Let $p^t = \{p_1^t, p_2^t, \ldots, p_N^t\}$ be the vector of task completion at stage $t$, where $p_i^t$ ($0 \leq p_i^t \leq 1$) is node $i$'s success ratio to fulfill its assigned task at stage $t$, and if there are $m$ active nodes $\{i_1, i_2, \ldots, i_m\}$ which are selected by the leader to cooperatively complete the key management task at this stage, then the expected system value will be

$$E(V^t) = \left( V^t \cdot \prod_{j=1}^{m} p_i^t \right).$$

3.2. Optimal Selection Model. We denote by $\mathcal{U}$ the class of all admissible nodes selection policies. The admissible policy $u \in \mathcal{U}$ is a $T \times N$ matrix, whose element of the $r$th row and the $i$th column is $a_i^r$, representing the action taken by node $i$ in stage $t$. The optimal nodes selection policy $u^r$ is the policy that achieves the system objective.

3.2.1. Cost. The total cost of system at stage $t$ is defined in (8), and the optimization objective is to find the optimal policy $u^r$ to minimize this cost as

$$Z^* = \min_{u^r \in \mathcal{U}} Z(u^r)$$

$$= \min_{u^r \in \mathcal{U}} \left[ \sum_{a_i^1=1}^{N} z_i^1(s_i^t, e_i^t, 1) + (N - m) \cdot C^t \right]$$

3.2.2. System Value. The expected value of system at stage $t$ is defined in (9), and the optimization objective is to find the optimal policy $u^r$ to maximize this value as

$$V^* = \max_{u^r \in \mathcal{U}} V(u^r) = \max_{u^r \in \mathcal{U}} \left[ V^r \cdot \prod_{a_i^r=1}^{m} p_i^r \right].$$
3.2.3. Optimal Nodes Selection Policy. Now, we have two important but conflicting objectives: minimizing the expected system cost (10) and maximizing the expected system value (11), and both have their own optimal policy $u^*$ and $u'^*$.

Hence, there is an intrinsic tradeoff between cost minimization and system value maximization. By introducing a new system parameter $\rho \in (0, 1)$ to combine these two objective functions together into a single objective function and using the weighting method [28], we formulate this multiobjective programming problem as a combinatorial optimization problem as follows:

$$u^* = \arg \max_{u \in \mathcal{U}} \left[ \rho \cdot V^* \cdot \prod_{i \in \mathcal{N}} p_i^* \cdot \frac{1}{1-\rho} \cdot \sum_{i \in \mathcal{N}} z_i^* \left( s_i^*, \xi_i^*, 1 \right) \right],$$

where the value of weight factor $\rho \in (0,1)$ can be set according to the application. For example, $\rho$ can be set to a value close to 1 in a battlefield MANET, in order to reflect the fact that the improvement of the success ratio of the key management task is more important than the reduction of the cost in the battlefield network. By contrast, in a civilian MANET, $\rho$ can be set to 0 to reflect the fact that the reduction of the costs is more important than the improvement of the success ratio of the key management task in the civilian network.

Since $\rho \in (0,1)$, the $1-\rho$ can be set to $0$ and we rewrite (12) as follows:

$$u^* = \arg \max_{u \in \mathcal{U}} \left[ \frac{\rho}{1-\rho} \cdot V^* \cdot \prod_{i \in \mathcal{N}} p_i^* \cdot \sum_{i \in \mathcal{N}} z_i^* \right].$$

Then the coefficient $(\rho/(1-\rho))$ can be omitted for simplicity, and we have

$$u^* = \arg \max_{u \in \mathcal{U}} \left[ W^* \cdot \prod_{i \in \mathcal{N}} p_i^* - \sum_{i \in \mathcal{N}} z_i^* \right].$$

By this we do not lose generality because when $\rho$ is given, we can substitute "$W^*"$ for $(\rho/(1-\rho)) \cdot V^*$, so that the coefficient of the new variation is equal to $1$.

4. Incentive Compatible Optimal Nodes Selection Mechanism Design

As stated before, mechanism design [13, 29] is concerned with the situation where a policy maker faces the problem of aggregating the individual preferences into a collective decision and the individuals’ actual preferences are not publicly known and studies how to elicit this privately held information and how the information revelation problem constrains the way in which social decisions can respond to individual preferences. To implement optimal nodes selection objective defined in (14), we apply game-theoretic approach to mechanism design and formulate the nodes selection process at each stage $t$ as a game where $N$ mobile nodes in the MANET are the players. Based on this model, we can design incentives to encourage each node in revealing its true information and honestly participate in the threshold key management process.

We assume that all nodes in the MANET are owned by rational and strategically selfish individuals, whose objectives are to maximize their individual goals. For this reason, these nodes may not always participate honestly in threshold key management, since this might cause security compromising and consume the nodes’ resources, including battery power, bandwidth, and CPU cycles. But as discussed in Section 3, the leader here can take the role of nodes selection and reputations payment loyalty. In this study, we just deal with the battery power consumption and security compromising, but our model can be extended to include more general cases straightforward.

4.1. The Mechanism. In each stage $t$, the leader initiates the game by asking each node, including itself, to reveal its type. Then, each node $i$ plays game by revealing its own private information based on its strategy $\xi_i^*$ drawn from an available strategy set $\Xi = \{\text{Truth}, \text{Untruth}\}$, according to how much the node values its utility with a utility function. If the node’s strategy is “Truth” then the node reveals the true type to the leader. If the node’s strategy is “Untruth”, then the node reveals a fake type to the leader. After receiving the revelations from each node, the leader takes these revelations as the input and makes its selection of $m$ nodes out of $N$ by using a given selection function. Each selected node is assigned a task to act as a server node and then cooperatively completes the current threshold key management task with other selected nodes. Finally, the leader pays the nodes by computing the payments vector $V^t = (V_1^t, \ldots, V_n^t)$ with a payment function. Payments are used to motivate nodes to behave in accordance with the mechanism goals.

In the rest of the paper, we use $\theta_i^t = (p_i^t, z_i^t)$ to denote the real type of node $i$ at stage $t$, and use superscript "w" to denote the type which is revealed to the leader, so to differentiate it from what is privately known by node itself. And use "−" to denote all the other nodes in nodes set $\mathcal{N}$ except $i$. Now we define ICONS mechanism which implements our optimal objective as follows.

4.1.1. Selection Function. Given the input of nodes’ revealed type vector $\theta^t$ at stage $t$, the mechanism of choosing $m$ nodes from nodes set $\mathcal{N}$ that maximizes the system’s welfare can be formulated as follows:

$$K_i^t \left( \hat{p}, \hat{z}^t \right) = \arg \max_{K_\mathcal{N}^t} \left( V^t \cdot \prod_{i \in K_\mathcal{N}^t} \hat{p}_i - \sum_{i \in K_\mathcal{N}^t} \hat{z}_i \right).$$

A branch and bound method [30] can be applied to allow us to find the optimal set of $m$ nodes $K_\mathcal{N}^t$ from MANET’s nodes set $\mathcal{N}$ as defined in (15), with reduced computational cost.

4.1.2. Payment Function. Let $q_i^t \in \{0, 1\}$ denote the node $i$’s completion of assigned key management task in stage $t$ if this node is active, where 1 means success in completing this task and 0 means fail. The payment will be given to each node $i$
by the leader in the form of reputation [31], according to the following payment function:

\[
R_i^t = \begin{cases} 
W^t \cdot \prod_{i \in K_i^t} \tilde{p}_i^t - K_i^t \left( \tilde{p}_i^t, z_i^t \right) - \sum_{i \in K_i^t} z_i^t & \text{node } i \text{ is active and } q_i^t = 1 \\
-K_i^t \left( \tilde{p}_i^t, z_i^t \right) - \sum_{i \in K_i^t} z_i^t & \text{node } i \text{ is active and } q_i^t = 0 \\
C_i^t & \text{node } i \text{ is passive.}
\end{cases}
\]

(16)

To let the leader detect if a selected node completes its task or not, we follow the previous work on developing an integrated fault-intrusion tolerance framework [32, 33] and do not differentiate between malicious faults and normal server failures (e.g., node crash, network disconnection, power failure, etc.) in our scenario. The detection method proposed in [34] that detects the corrupted shares for the proactive secret sharing can be adopted here, by checking if the node participates in the process with the presence of an uncorrupted share of system’s secret. A recent similar work is found in [35, 36], which uses Shamir’s secret sharing scheme to detect malicious activities in the encrypted networks such as virtual private networks (VPNs) that encrypt and conceal network traffic.

Now, we get the utility function for node \( i \) at stage \( t \) as follows:

\[
u_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) = R_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t \right) - z_i^t,
\]

(17)

where \( z_i^t \) is the instantaneous cost of node \( i \) at stage \( t \), as defined in (7); \( R_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t \right) \) is the payment defined in (16) given by the mechanism to the node \( i \), when the coalition of \( m \) nodes selected from \( N \) to cooperatively complete the key management task at stage \( t \) is \( K_i^t \); \( \tilde{\theta}_i^t \) and \( \tilde{\theta}_{-i}^t \) are revealed type(s) of node \( i \) and nodes set \( \{-i\} \), respectively.

Then, node \( i \)'s expected utility at stage \( t \) is

\[
u_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) = E \left[ u_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) \right].
\]

(18)

Note that \( \nu_i^t \left( K_i^t, \tilde{\theta}_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) \) is what the node \( i \) usually seeks to maximize. It reflects the amount of benefits gained by node \( i \) if it follows a specific strategy \( \tilde{\theta}_i^t \) at stage \( t \). Nodes might deviate from revealing their truthful types if that could lead to a better payment. Therefore, our mechanism must be strategyproof where truth revealing is the dominant strategy, and thus the following standard properties are required to be satisfied [13, 29].

1. Individual rationality holds when truthful nodes are guaranteed to have nonnegative expected utility. Formally, this condition holds, when for all \( i, \theta_i^t, \) and \( \tilde{\theta}_{-i}^t \),

\[
u_i^t \left( K_i^t, \theta_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) \geq 0.
\]

(19)

2. Incentive compatibility holds when it is a dominant strategy for each node with truthful revelation. Formally, for all \( i, \theta_i^t, \theta_{i}^t, \) and \( \tilde{\theta}_{-i}^t \),

\[
u_i^t \left( K_i^t, \theta_i^t, \tilde{\theta}_{-i}^t, z_i^t \right) \geq \nu_i^t \left( K_i^t, \theta_{i}^t, \tilde{\theta}_{-i}^t, z_i^t \right).
\]

(20)

3. No-free-riders holds if all nodes not selected to participate in the current key management have a revenue of 0.

The properties of individual rationality, incentive compatibility, and no free riders imply that, (1) the expected utility of a truthful node is always nonnegative; (2) each node will find no better option than to reveal their true type; (3) the nodes that are not selected to participate in the current key management have a revenue of 0. Therefore, all rational nodes that include the malicious ones will find that revealing their types untruthfully can never lead to a better payment than revealing their types truthfully, and sending no information to the leader can never lead to a better payment than reporting their types to the leader. Then all rational nodes will always report their types truthfully to the leader, since their objectives are to maximize their individual benefits.

Similarly in our mechanism, leader also need to maximize the system’s welfare, and so we have the following definition.

Definition 1. A selection function is called socially efficient if the chosen selection \( K^t \in \mathcal{N}^m \) maximizes social welfare over \( \mathcal{N}^m \); that is, for all \( K^t \in \mathcal{N}^m \),

\[
\left( W^t \cdot \prod_{i \in K^t} \tilde{p}_i^t - \sum_{i \in K^t} z_i^t \right) \geq \left( W^t \cdot \prod_{i \in K^t} \tilde{p}_i^t - \sum_{i \in K^t} z_i^t \right).
\]

(21)

Now, we have the following required property.

4. A mechanism is called a socially efficient mechanism, if it has a socially efficient selection function.

To ensure truth elicitation from all the nodes, we need to prove that the presented mechanism is strategyproof.

4.2. Properties of Mechanism

4.2.1. Individual Rationality. Individual rationality means that the expected utility of a truthful node is always nonnegative. Truthful node \( i \) with its revealed type \( \theta_i^t \) might either be selected to participate in completing the key management task at stage \( t \) or not, given other nodes \( -i \)’s revealed types vector \( \tilde{\theta}_{-i}^t \). Now, we consider both cases as follows.

Case 1. Truthful node \( i \) is not selected to participate in the key management at stage \( t \).

From (18) and (16), we know that node \( i \)'s expected utility is 0 at this stage, because both its payment and its cost are \( C_i^t \) and hence we proved our claim.

Case 2. Truthful node \( i \) is selected to participate in the key management task at stage \( t \).

From (18) and (16), we know that node \( i \)'s expected utility in this case is

\[
u_i^t = R_i^t - z_i^t
\]

\[
= p_i^t \left( W^t \cdot \prod_{j \in K_i^t} \tilde{p}_j^t - K_i^t \left( \tilde{p}_j^t, z_j^t \right) - \sum_{j \in K_i^t} z_j^t \right)
\]

(20)
node is truthful at this stage, we have $\hat{\theta}_i = \theta_i$, that means $\bar{p}_i = p_i$ and $\bar{z}_i = z_i$. Therefore, (22) can be rewritten as

$$
\bar{u}_i = p_i \cdot W_t \cdot \prod_{j \in K_i^r, j \neq i} \bar{p}_j - \mathcal{K}_{i,i}^t (\bar{p}_i^t, \bar{z}_i^t) - \sum_{j \in K_i^r, j \neq i} \bar{z}_j^t - z_i^t.
$$

(22)

Since node $i$ is truthful at this stage, we have $\hat{\theta}_i = \theta_i$, that means $\bar{p}_i = p_i$ and $\bar{z}_i = z_i$. Therefore, (22) can be rewritten as

$$
\bar{u}_i = p_i \cdot W_t \cdot \prod_{j \in K_i^r, j \neq i} \bar{p}_j - \mathcal{K}_{i,i}^t (\bar{p}_i^t, \bar{z}_i^t) - \sum_{j \in K_i^r, j \neq i} \bar{z}_j^t - z_i^t.
$$

(23)

According to the selection function defined in our mechanism, the first term in (23) quantifies the optimal welfare that can be obtained when node $i$'s revealed type is its true one $\bar{\theta}_i$, the vector of other nodes' revealed types is $\bar{\theta}_{-i}$, and node $i$ is involved in the selection. Similarly, the second term quantifies the optimal welfare that can be obtained when the vector of other nodes' revealed types is $\hat{\theta}_{-i}$, but node $i$ is not involved in the selection. Since node $i$'s involvement can only improve the total welfare, we have $\bar{u}_i = K^i_\theta (\bar{p}, \bar{z}) - K_{-i} (\bar{p}_i, \bar{z}_i) \geq 0$ and proved this property.

4.2.2. Incentive Compatibility. Incentive compatibility means that players will find no better option than to reveal their true type. We consider the node $i$ and other nodes $-i$. Given revealed types of $i$ and $-i$, $\bar{\theta}_i = \{\bar{p}_i, \bar{z}_i\}$ and $\bar{\theta}_{-i} = \{\bar{p}_{-i}, \bar{z}_{-i}\}$, we need to show that node $i$ cannot gain more from not revealing its true type than revealing its true type. Now given its true type $\theta_i$ at stage $t$, we consider two cases of this node by revealing its true types, namely, either selected or not.

Case 1. Node $i$ is selected by the leader if it reveals its type $\theta$ truthfully at stage $t$. Then from the property of individual rationality, we know that by revealing its true type at this case, node $i$ can gain a utility $\bar{u}_i \geq 0$. Now we consider two subcases of node $i$'s untruthful revelation.

(i) If it is not selected by the leader due to its untruthful revelation, then node $i$'s utility at stage $t$ will still be 0, and this make node $i$ have no incentive to be untruthful at this subcase.

(ii) If it is selected by the leader to participate in completing the key management task at current stage, then from (22) and the payment function of our mechanism we know that, given other nodes’ revealed types $\bar{\theta}_{-i}$, node $i$’s expected utility at current stage just relates to node $i$’s true type $\theta_i$, no matter what type it had revealed. That is to say, untruthful revelation cannot make extra utility to node $i$, so node $i$ has no incentive to be untruthful at this subcase also.

Case 2. Node $i$ is not selected to participate in the key management task at stage $t$. From (16), we know that by revealing its true type, node $i$ could gain a utility of 0 at this case. Now we consider two subcases of node $i$ with an untruthful revelation.

(i) Node $i$ is still not selected to participate in the task with its untruthful revelation.

In this subcase, node $i$’s utility is still 0, and this means that the utility of node $i$ remains the same with an untruthful revelation of type $\hat{\theta}_i$.

(ii) Node $i$ is selected to participate in the task because of its untruthful revelation. From (22) and the payment function in our mechanism, we know that node $i$’s expected utility $u_i^n$ at this scenario will be

$$
u_i^n = \left[ p_i^t \cdot W_t \cdot \prod_{j \in K_i^r, j \neq i} \bar{p}_j - \left( \sum_{j \in K_i^r, j \neq i} z_j^t + z_i^t \right) \right] - K_{-i} (\bar{p}_i, \bar{z}_i) - K_{-i} (\bar{p}_i, \bar{z}_i) \]

(24)

where $(\bar{p}_i, z_i^t)$ is node $i$’s true type at stage $t$, and $K^t_\theta (p_i, \bar{p}_{-i}, \bar{z}_{-i})$ is a new selection scheme over the vector of nodes’ revealed types $(\theta_i, \bar{\theta}_{-i})$, with node $i$ being selected to participate in completing the task at stage $t$, that is, $i \in K_i^t$.

Note that in this subcase, node $i$ would not be selected by the leader if it revealed its true type, and so with its truthful revelation, node $i$ whether involved in the selection or not will make no difference on system’s welfare at this stage. Then we have

$$
K^t_\theta (\bar{\theta}_i, \bar{\theta}_{-i}) = K^t_\theta (\bar{p}_i, \bar{z}_i) = K_{-i} (\bar{p}_i, \bar{z}_i).
$$

(25)

If we assume $u_i^n > 0$ and substitute $K_{-i} (\bar{p}_i, \bar{z}_i)$ with $K^t_\theta (p_i, \bar{p}_{-i}, \bar{z}_{-i})$ in (24), then we can get

$$
u_i^n = K^t_\theta (p_i, \bar{p}_{-i}, \bar{z}_{-i}) - K_{-i} (\bar{p}_i, \bar{z}_i) > 0,
$$

(26)

where $K^t_\theta (p_i, \bar{p}_{-i}, \bar{z}_{-i})$ is the optimal nodes selection scheme over the vector of nodes’ revealed types $(\theta_i, \bar{\theta}_{-i})$ at
stage $t$. This means that, given the revelation of node types $(\hat{\theta}_i, \hat{\theta}_{-i})$, new selection scheme $K_{s^*}^i$ is more optimal than $K_{s}^i$. However, this contradicts the definition of $K_{s^*}^i$. Therefore, revealing an untruthful type $\hat{\theta}_i$ cannot lead to a higher utility of node $i$ in this subcase.

4.2.3. No Free Riders. This property can be derived from the payment function of our mechanism directly.

4.2.4. Socially Efficient. This property can be derived from incentive compatibility and the selection function of our mechanism directly.

5. Simulation Experiments and Results

In this section, we illustrate some of the performance benefits of our proposed model. To show efficient improvement of our model and to show the negative impact of selfish node with untruth telling, we evaluate the performance of our strategyproof optimal nodes selection model with respect to Yu’s selection model [10] and random selection model [2–9].

To eliminate the effect of leader election phase’s cost, a reasonable choice would be to run the leader election algorithm only once and follow a fixed number of stages after the initial election and thus amortizing the overhead through the many iterations of the key management tasks, similar to what is explained in [37]. In this way, the actual overhead of leader election would be neglected when considering the network lifetime.

For simplicity, we use three security states: safe ($s$), vulnerable ($v$), and compromised ($c$) and three energy states high ($b_1$), middle ($b_2$), and low ($b_3$). There are a total of nine states for a node $(s, b_1), (s, b_2), (s, b_3), (v, b_1), (v, b_2), (v, b_3), (c, b_1), (c, b_2),$ and $(c, b_3)$ as defined in (3). The state transition probability matrix stands for the probability of a node changes from one state to another. We define the basic security state transition probability matrix $A_1$ and basic energy state transition probability matrix $B_1$ for active nodes as follows: $A_1 = \begin{bmatrix} 0.92, 0.05, 0.03, 0.04, 0.92, 0.04, 0.01, 0.03, 0.96; 0.94, 0.04, 0.02, 0.02, 0.03, 0.95, 0.01, 0.03, 0.96; 0.93, 0.05, 0.02, 0.03, 0.94, 0.03, 0.03, 0.02, 0.95; 0.998, 0.001, 0.001, 0.001, 0.0998, 0.001, 0.001, 0.001, 0.0999 \end{bmatrix}$ and $B_1 = \begin{bmatrix} 0.98, 0.02, 0.00, 0.00, 0.98, 0.02, 0.00, 0.00, 1.00; 0.99, 0.01, 0.00, 0.00, 0.99, 0.01, 0.00, 0.00, 1.00. \end{bmatrix}$ In our simulation, each active node can choose its security state transition probability matrix and energy state transition probability matrix from $A_1$ and $B_1$ with an initialization function randominit. For example, when node $i$, with a security state of vulnerable and a energy state of middle in current stage, chooses $[0.93, 0.05, 0.02, 0.03, 0.94, 0.03, 0.03, 0.02, 0.95]$ and $[0.99, 0.01, 0.00, 0.00, 0.99, 0.01, 0.00, 0.00, 1.00]$ as its security state transition probability matrix and energy state transition probability matrix, respectively, then its security state in the next stage will be compromised with probability 0.30, will remain be vulnerable with probability 0.94, and will be snatched back to the safe state with probability 0.30; its energy state in the next stage will still be middle with probability 0.99 but low with probability 0.01. We also assume that when a node is passive, the transition probability is lower than that when the node is active, and so we define the basic security state transition probability matrix $A_2$ and basic energy state transition probability matrix $B_2$ for passive nodes as follows: $A_2 = \begin{bmatrix} 0.99, 0.01, 0.00, 0.99, 0.01, 0.00, 0.99, 0.01, 0.00 \end{bmatrix}$ and $B_2 = \begin{bmatrix} 0.99, 0.01, 0.00, 0.99, 0.01, 0.00, 0.99, 0.01, 0.00 \end{bmatrix}$.

To conduct parameter-sensitivity analysis and check the impacts of various transition probabilities on the performance in the simulations, we change the transition probability from 0.98 to 0.82 with the matrix of $[0.98, 0.02, 0.00, 0.00, 0.98, 0.02, 0.00, 0.00, 1.00; 0.96, 0.04, 0.00, 0.00, 0.96, 0.04, 0.00, 0.00, 0.94, 0.06, 0.00, 0.00, 0.94, 0.06, 0.00, 0.00, 1.00; 0.92, 0.08, 0.00, 0.00, 0.92, 0.08, 0.00, 0.00, 0.90, 0.10, 0.00, 0.00, 0.90, 0.10, 0.00, 0.00, 1.00; 0.88, 0.12, 0.00, 0.00, 0.88, 0.12, 0.00, 0.00, 0.86, 0.14, 0.00, 0.00, 0.86, 0.14, 0.00, 0.00, 1.00; 0.84, 0.16, 0.00, 0.00, 0.84, 0.16, 0.00, 0.00, 0.84, 0.16, 0.00, 0.00, 0.84, 0.16, 0.00, 0.00, 1.00; 0.82, 0.18, 0.00, 0.00, 0.82, 0.18, 0.00, 0.00, 0.82, 0.18, 0.00, 0.00, 0.82, 0.18, 0.00, 0.00, 1.00]. Since the risk of damage to the network would be further increased if a compromised node is chosen to act as a server node, we set the cost of selecting a higher security state to lower values than that associated with a lower security state selection. Similarly, to balance the nodes’ energy consumption and avoid the situation where there are some higher energy level nodes still in a dead network, the selection of a lower energy level node should have a higher cost than selecting a higher energy level node. Thus, we define the cost matrices for the simulation as follows: $C_1 = [5.5, 7.5, 11, 25, 31, 37, 43, 52, 63], C_2 = [6, 8, 11, 23, 30, 35, 42, 50, 62], C_3 = [6.5, 8.5, 12, 21, 28, 33, 40, 48, 60], C_4 = [8, 10, 13, 18, 25, 30, 37, 45, 55], C_5 = [8.5, 11, 15, 19, 27, 32, 38, 47, 58], C_6 = [9, 10, 15, 20, 28, 32, 38, 48, 59], C_7 = [9, 12, 16, 15, 20, 28, 33, 40, 50], C_8 = [10, 13, 18, 16, 21, 28, 34, 42, 51], C_9 = [11, 14, 19, 17, 22, 29, 35, 43, 52],$ corresponding to the system state matrix $(s, b_1), (s, b_2), (s, b_3), (v, b_1), (v, b_2), (v, b_3), (c, b_1), (c, b_2), (c, b_3)$. The system value of the threshold key management service in our simulation is defined as $V^t = 300, \rho = 0.25$, and then $W^t = 200$ for each stage.

5.1. Cost Reduction. First, we compare the costs in different models along simulation stages when a $(3, 7)$ secret sharing scheme is used. Initializing each node with a state of $(1, 1)$, we can see from Figure 1 that there is a distinct cost reduction of both our strategyproof optimal selection model and Yu’s selection model over the random selection model. This is mainly due to the fact that the random selection model selects nodes without considering the cost and thus leading to a higher average cost. Figure 1 also demonstrates that for the two optimal selection models, there is a better performance of our selection model than that of Yu’s selection model. This result indicates that, with the presence of selfish nodes, the normal nodes in the system must more often be active to carry out the duty of key management than the nodes in the system without selfish nodes and so will transfer into more cost states with a higher probability.

We also perform parameter-sensitivity analysis on the models by considering different crypto thresholds. The simulation is performed with 30 stages for 200 times and then the average costs of each stage are calculated. Figure 2 shows the costs comparison over our model, the random selection model.
model, and Yu's selection model, when there are 15 nodes participating in the key management service, all with initial node state of (1, 1), with the crypto threshold changing from 2 to 7. With the increase of the crypto threshold, the cost to perform the key management task increases due to more nodes that need to be active, but our model always has the lowest cost.

5.2. Network Lifetime Improvement. In these simulations, we investigate the network lifetime improvement of the proposed model. Let $N$ be the number of nodes in the MANET, and we consider the crypto threshold as $(N - 1)/2$ in the simulation. So when there are $N - (N - 1)/2 = (N + 1)/2$ nodes run out of power, the network is regarded as dead. A node is considered running out of power, if it has run $x$ stages on passive mode and $y$ stages on active mode since entering into the low ($h3$) energy state, where $x + 2y = 30$ and $0 \leq x, y \leq 30$. We first check the performance when different energy transition probabilities are used. We set $N = 7$ and let the energy transition probability of the active node be in the range from 0.88 to 0.98. The energy transition probabilities matrix for passive node remains unchanged in all circumstance as defined before, that is, $B2 = [0.99, 0.01, 0, 0.99, 0.01, 0, 0, 1]$. As shown in Figure 3, our strategyproof optimal selection model and Yu's selection model always have longer network lifetime than the random selection model, because in random selection model the nodes are selected without considering the energy level and this leads the nodes with low energy to die fast.

Then we check the performance when different numbers of nodes are available in the network. From Figure 4 we can see that the key management service is distributed among more nodes and thus prolonging the lifetime of the network in all three models (random selection model, Yu's selection model, and our selection model), with the number of available nodes in MANET increasing from 5 to 35. Still the same as before, our optimal selection model shows consistent improvement over the other two models in this simulation.

5.3. Success Ratio Improvement. The success ratio is the probability that the leader successfully collects all the requested partial signature or partial authentication from a threshold number of nodes and then completes the assigned key management task. We assume that a node in safe state will always complete its assigned task with probability of 1, while a vulnerable node and a compromised node with probabilities of $1/\sqrt{2}$ and $1/\sqrt{3}$, respectively. First, we compare the average success ratios of threshold key management in different models along simulation stages when there are 7 nodes in the network, with the crypto threshold and selfish nodes number set to 3 and 2, respectively. Figure 5 shows that our strategyproof optimal selection model has a distinct success ratio improvement over the random selection model and also demonstrates a better success ratio than Yu's selection model. This is mainly due to the fact that neither the random selection model nor Yu's selection model selects nodes with the consideration of the success ratios of each nodes.

We then show the success ratio improvement when there are more nodes in the network. From Figure 6 we can see, with the number of available nodes in the network increasing from 7 to 19, that the success ratios of all three models become higher since there are more nodes with higher security state which can be selected from. The success ratio of our optimal selection model still is shown to be the highest in these circumstances.

5.4. Network Compromising Probability Reduction. Last but not least, we investigate the probability of the network being compromised by attacker(s) who is(are) attempting to assemble enough key information to deduce the master key of the system. In order to quantify and compare different models in our scenario, we will use as a metric the compromise probability that is defined as the probability that an attacker can recover the master key of the secret sharing scheme, after capturing enough nodes, and so is inversely proportional to the number of stages required by attacker(s) to capture the required number of nodes and then to compromise the network. Assume that the attacker knows all public parameters of the system, then when a $(m, N)$ secret sharing
scheme is used, the MANET is deemed compromised if \((N - m)\) nodes are captured by the attacker(s). In our simulation, a node is defined as captured, if it has run \(x\) stages on passive mode and \(y\) stages on active mode since it entered into the compromised security state, where \(x + 2y = 30\) and \(0 \leq x, y \leq 30\).

Firstly, we set \(N = 7, m = 3\), and compare the network compromising probabilities when security transition probabilities are in the range from 0.76 to 0.98. The security transition probabilities matrix for passive node remain unchanged in all circumstance as defined before, that is, \(A_2 = [0.99, 0.01, 0, 0, 0.99, 0.01, 0.005, 0.005, 0.99; 0.999, 0.001, 0, 0, 0.999, 0.001, 0, 0.001, 0.999, 0.001, 0, 0.001, 0.999]\). Among the 7 nodes, we assume in the beginning, that 3 are in safe security state, 2 in vulnerable security state that, and 2 in compromised security state.

The results in Figure 7 indicate that the proposed selection model has lower network compromising probability than the random selection model and Yu’s selection model, since our strategyproof optimal selection model tends to select nodes with higher security levels and thus keeping a balance of all node’s security level with time. When the transition probabilities are closer to 1, the compromising probabilities of all models asymptotically approach 0. This is because the nodes in each model will keep their security state unchanged and so keep the networks safe in each stage. In Figure 8, we compare the network compromising probability when there are different numbers of nodes in the network. With an increase in the total number of nodes in the MANET, all the models show a downward trend in compromising probabilities because the key management service can be distributed among more nodes which will decrease the probability of each node transition into a lower security level. Similarly as before, our optimal selection model has lower compromising probability than the other two models.

From the simulation results in Figures 1–8, we can see that with the incentive compatible optimal nodes selection mechanism to encourage each node in the system to be truth-telling and to select a coalition of nodes with the purposes of maximizing the success ratio of key management service...
and minimizing the nodes’ cost, our optimal nodes selection model certainly has dramatic results in reducing the nodes’ resource consumption and network’s security compromising, while improving the lifetime of the network and the success ratio of the key management service dramatically, in the presence of selfish nodes.

6. Conclusion and Discussion

In this paper, focusing on the optimal nodes selection problem in presence of selfish nodes for threshold key management in MANETs during its operation, we formulated the dynamic nodes selection problem as a combinatorial optimization problem firstly, with the objectives of maximizing the success ratio of key management service and minimizing the nodes’ cost of security and energy and then proposed the incentive compatible mechanism to implement the optimal nodes selection process in MANETs, to ensure that the truth telling is the dominant strategy and so prevent the emergence of selfish nodes. To the best of our knowledge, this is the first incentive-compatible mechanism for threshold key management.

In our scheme, although one of the nodes in the network needs to be specified as a leader, essentially there are differences between the leader node in this scheme and the PKG server in the centralized scheme [38, 39], and these make our proposed mechanism efficient and suitable for the MANETs. Specifically, (1) instead of being specified by the administrator in bootstrapping network phase and fixed during network lifetime, the leader can be elected/relected periodically and/or when found to be failed, attacked, or run out of battery, it cannot reach any nodes in the system, and so forth and so eliminating the single point of failure. (2) The network’s primary secret is not held by the leader itself but is split and shared by all the nodes by using secret sharing method. Therefore, no node in the network is required to be trusted and available to all other nodes. (3) The main task of key management is not performed by the leader but is performed by a threshold number of nodes selected at each stage elaborately, with the consideration of maximizing the success ratio of key management service and minimizing the nodes’ cost of security and energy. In this way, our scheme improves both the success ratio of key management service and lifetime of the network and reduces both the cost of participating nodes and compromising probability. (4) The ICONS mechanism that we proposed cannot only be used in the scenario of threshold key management but also in other cooperation scenarios in MANET.

For future work, we plan to extend this mechanism to a distributed setting [40]. Although we argued in Section 3 that it is reasonable to assume that there is always a node in the network to act as a leader, a practical and distributed selection model without a specific leader node will be more helpful to implement the nodes selection model in the real world.

Acknowledgments

This work was supported by the Excellent Youth Foundation of Henan Scientific Committee (104100510025), the National Natural Science Foundation of China (no. 60633020), and the Open Foundation from the Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education (201107).

References

[1] S. Capkun, J. P. Hubaux, and L. Buttyan, “Mobility helps peer-to-peer security,” IEEE Transactions on Mobile Computing, vol. 5, no. 1, pp. 43–51, 2006.
[2] L. Zhou and Z. Haas, “Securing ad hoc networks,” IEEE Network, vol. 13, pp. 24–30, 1999.
[3] B. Zhu, F. Bao, R. H. Deng, M. S. Kankanhalli, and G. Wang, “Efficient and robust key management for large mobile ad hoc networks,” Computer Networks, vol. 48, no. 4, pp. 657–682, 2005.
[4] K. Hamoudi and K. Adi, “Secure and robust threshold key management (SRKM) scheme for ad hoc networks,” Security and Communication Networks, vol. 3, Article ID 517534, 2010.
[5] Y. Dong, A. F. Sui, S. M. Yiu, V. O. K. Li, and L. C. K. Hui, “Providing distributed certificate authority service in cluster-based mobile ad hoc networks,” Computer Communications, vol. 30, pp. 2442–2452, 2007.

[6] D. Y. Lee and H. C. Jeong, “An efficient certificate management for mobile ad-hoc network,” in Ad-Hoc, Mobile, and Wireless Networks, T. Kunz and S. Ravi, Eds., vol. 4104 of Lecture Notes in Computer Science, pp. 355–364, Springer, Berlin, Germany, 2006.

[7] S. Yi and R. H. Kravets, “MOCA: mobile certificate authority for wireless ad hoc networks,” in Proceedings of the 2nd Annual PKI Research Workshop (PKI ’03), 2003.

[8] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, “URSA: ubiquitous and robust access control for mobile ad hoc networks,” IEEE/ACM Transactions on Networking, vol. 12, no. 6, pp. 1049–1063, 2004.

[9] M. Omar, Y. Challal, and A. Bouabdallah, “Reliable and fully distributed trust model for mobile ad hoc networks,” Computers and Security, vol. 28, no. 3–4, pp. 199–214, 2009.

[10] F. R. Yu and H. Tang, “Distributed node selection for threshold key management with intrusion detection in mobile ad hoc networks,” Wireless Networks, vol. 16, no. 8, pp. 2169–2178, 2010.

[11] V. A. Leoni and G. L. Nasini, “On the computational complexity of combinatorial flexibility problems,” International Journal of Computer Mathematics, vol. 87, no. 15, pp. 3330–3343, 2010.

[12] L. Hurwicz and S. Reiter, Designing Economic Mechanisms, Cambridge University Press, Cambridge, UK, 4th edition, 2008.

[13] A. Mas-Colell, M. D. Whinston, and J. R. Green, Microeconomic Theory, Oxford University Press, Oxford, UK, 1995.

[14] T. Wu, M. Malkin, and D. Boneh, “Building intrusion tolerant applications,” in Proceedings of the 8th Conference on USENIX Security Symposium, vol. 8, p. 7, USENIX Association, 1999.

[15] M. Masdari, S. Jabbehdari, M. R. Ahmadi, S. M. Hashemi, J. Bagherzadeh, and A. Khadem-Zadeh, “A survey and taxonomy of distributed certificate authorities in mobile ad hoc networks,” EURASIP Journal on Wireless Communications and Networking, vol. 2011, pp. 1–12, 2011.

[16] J. Sen, M. Girish Chandra, P. Balamuralidhar, S. G. Harihara, and H. Reddy, “A scheme of certificate authority for ad hoc networks,” in Proceedings of the 18th International Workshop on Database and Expert Systems Applications (DEXA ’07), pp. 615–619, September 2007.

[17] D. Joshi, K. Namuduri, and R. Pendse, “Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: an analysis,” EURASIP Journal on Wireless Communications and Networking, vol. 2005, no. 4, pp. 579–589, 2005.

[18] L. Anderegg and S. Eidenbenz, “Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile Ad hoc networks with selfish agents,” in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom ’03), pp. 245–259, September 2003.

[19] Y. Wang and M. Singhal, “On improving the efficiency of truthful routing in MANETs with selfish nodes,” Pervasive and Mobile Computing, vol. 3, no. 5, pp. 537–559, 2007.

[20] A. Rachedi, A. Benslimane, H. Otrok, N. Mohammed, and M. Debbari, “A secure mechanism design-based and game theoretical model for MANETs,” Mobile Networks and Applications, vol. 15, no. 2, pp. 191–204, 2010.

[21] D. Hao, Y. Ren, and K. Sakurai, “A game theory-based surveillance mechanism against suspicious insiders in MANETs,” in Trusted Systems, vol. 6802 of Lecture Notes in Computer Science, pp. 237–252, Springer, Berlin, Germany, 2011.

[22] N. Mohammed, H. Otrok, L. Wang, M. Debbari, and P. Bhattacharya, “Mechanism design-based secure leader election model for intrusion detection in MANET,” IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 1, pp. 89–103, 2011.

[23] H. Garcia-Molina, “Elections in a distributed computing system,” IEEE Transactions on Computers, vol. 31, pp. 48–59, 1982.

[24] H. C. Chung, P. Robinson, and J. L. Welch, “Optimal regional consecutive leader election in mobile ad-hoc networks,” in Proceedings of the 7th ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing (FOMC ’11), pp. 52–61, June 2011.

[25] Q. Dong and D. Liu, “Resilient cluster leader election for wireless sensor networks,” in Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’09), pp. 108–116, IEEE Press, June 2009.

[26] P. Hu, Z. Zhou, Q. Liu, and F. Li, “The HMM-based modeling for the energy level prediction in wireless sensor networks,” in Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications, pp. 2253–2258, May 2007.

[27] P. Samarati, M. K. Reiter, and S. Jajodia, “An authorization model for a public key management service,” ACM Transactions on Information and System Security, vol. 4, pp. 453–482, 2001.

[28] K. Miettinen, Non Linear Multi-Objective Optimization, Kluwer Academic Publishers, Norwell, Mass, USA, 1999.

[29] N. Nisan, “Algorithms for selfish agents mechanism design for distributed computation,” in Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS ’99), pp. 1–15, Springer, Trier, Germany, 1999.

[30] M. J. Brusco and S. Stahl, Branch-and-Bound Applications in Combinatorial Data Analysis, Springer, New York, NY, USA, 2005.

[31] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, “Reputation systems: facilitating trust in internet interactions,” Communications of the ACM, vol. 43, no. 12, pp. 45–48, 2000.

[32] A. Bessani, “From byzantine fault tolerance to intrusion tolerance,” in Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSNW ’11), pp. 15–18, 2011.

[33] Q. Nguyen and A. Sood, “A comparison of intrusion-tolerant system architectures,” IEEE Security Privacy, vol. 9, pp. 24–31, 2011.

[34] A. Herzberg, S. Jarecki, K. Krawczyk, and M. Yung, “Proactive secret sharing or: how to cope with perpetual leakage,” in Proceedings of the 15th Annual International Cryptology Conference Advances in Cryptology (CRYPTO ’95), pp. 339–352, Springer, Santa Barbara, Calif, USA, 1995.

[35] V. T. Goh, Intrusion detection framework for encrypted networks [Ph.D. thesis], Queensland University of Technology, Brisbane, Australia, 2010.

[36] V. T. Goh, J. Zimmermann, and M. Looi, “Experimenting with an intrusion detection system for encrypted networks,” International Journal of Business Intelligence and Data Mining, vol. 5, pp. 172–191, 2010.

[37] M. Conti, R. Di Pietro, L. Mancini, and A. Mei, “Distributed detection of clone attacks in wireless sensor networks,” IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 5, pp. 685–698, 2011.

[38] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt, and S. Wolthusen, “Strongly-resilient and non-interactive hierarchical key-agreement in MANETs,” in Proceedings of the 13th European Symposium on Research in Computer Security, pp. 49–65, Springer, 2008.
[39] H. Deng, A. Mukherjee, and D. P. Agrawal, “Threshold and identity-based key management and authentication for wireless ad hoc networks,” in Proceedings of the International Conference on Information Technology: Coding and Computing, pp. 107–111, IEEE Computer Society, 2004.

[40] D. Monderer and M. Tennenholtz, “Distributed games: from mechanisms to protocols,” in Proceedings of the 16th National Conference on Artificial Intelligence (AAAI ’99), pp. 32–37, American Association for Artificial Intelligence, 1999.