Key Revocation Based on Dirichlet Multinomial Model for Mobile Ad Hoc Networks

Xinxin Fan, and Guang Gong, *Member, IEEE*

Abstract— **The absence of an online trusted authority makes the issue of key revocation in mobile ad hoc networks (MANETs) particularly challenging. In this paper, we present a novel self-organized key revocation scheme based on the Dirichlet multinomial model and identity-based cryptography (IBC). Our key revocation scheme offers a theoretically sound basis for a node in MANETs to predict the behavior of other nodes based on its own observations and reports from peers. In our scheme, each node keeps track of three categories of behavior defined and classified by an external trusted authority, and updates its knowledge about other nodes' behavior with 3-dimension Dirichlet distribution. Differentiating between suspicious behavior and malicious behavior enables nodes to make multilevel response by either revoking keys of malicious nodes or ceasing the communication with suspicious nodes for some time to gather more information for making further decision. Furthermore, we also analyze the attack-resistant properties of our key revocation scheme through extensive simulations in the presence of adversaries.**

Index Terms— **Mobile ad hoc networks, security, key revocation, identity-based cryptography, Dirichlet multinomial model.**

I. Introduction

Mobile ad hoc networks (MANETs) provide a relative new paradigm of wireless networking, in which all networking functions (e.g., control, routing, monitoring, mobility management, etc.) are performed by the nodes themselves in a decentralized manner. Security support is indispensable in order for these networks and related services to be implemented in both military and commercial applications. However, due to the absence of infrastructure, insecure nature of the wireless communication medium and dynamic changes of the network topology, MANETs are vulnerable to a range of attacks and are thus difficult to secure [1], [4].

In this paper, we address the key revocation, one of the most important and challenging issues for the key management in MANETs. Conventional techniques for revocation using certificate revocation lists (CRLs) and certification authorities (CAs) are difficult to be applied to MANETs because of a number of unique features of MANETs such as the absence of an on-line CA and a centralized repository. Two categories of solutions have been proposed for the key revocation in MANETs and each of them can be implemented with the certificate-based cryptography (CBC) or identity-based cryptography (IBC). The first category of solutions use threshold cryptography and some network nodes collaborate to revoke keys of malicious nodes, whereas the second category of solutions are fully self-organized, in which each node has its own view about the network and decides whether keys of other nodes should be revoked based on its own observations and the information collected from peers in MANETs.

Previous key revocation schemes for MANETs all classify the behavior of nodes as either *good* or *bad* without any intermediate state. Such a binary behavior differentiation omits the actual cause and the degree of misbehavior. Note that some misbehavior may just happen accidently (for example, a node cannot forward packages due to temporary congestion of the network) and last only for a short time. When a node shows this kind of accidental misbehavior, it might not mean that the node has been compromised by an attacker. Therefore, in this case it is more reasonable to keep collecting information about the behavior of this node instead of immediately characterizing it as malicious and excluding it from the network. To provide more flexibility and precision for nodes analyzing peers' behavior and making different response based on results of the analysis, we present a novel self-organized key revocation scheme based on IBC and Dirichlet multinomial model in this contribution. By differentiating multi-categories of misbehavior based on their actual cause, we hope that keys of good nodes who only misbehave for a short time due to various reasons will not be immediately revoked by other good nodes. However, the more categories of malicious behavior, the more complicated the implementation. Thus, in this paper we only consider two categories of misbehavior, namely suspicious behavior and malicious behavior. To establish multi-parameter Bayesian model for analyzing nodes' behavior, we employ Dirichlet reputation systems proposed by Jøsang and Haller [12] and make some modifications about the information integration based on Dampster-Shafer belief theory [18].

The rest of this paper is organized as follows: Section II gives a short introduction to mathematical tools used in this paper. Section III formulates the network and security models. Section IV describes our key revocation scheme, followed by simulations and analysis of our scheme in Section V. Section VI reviews existing solutions related to our work. This paper is finally concluded in Section VII.

II. Preliminaries

A. IBC and Bilinear Pairing

The concept of IBC is due to Shamir [19]. In an ID-based cryptosystem, a user's public key is an easily calculated function of his identity, while his private key can be computed by a TTP. Many interesting ID-based cryptographic protocols use so-called bilinear pairings which are defined as follows.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: x5fan@engmail.uwaterloo.ca, ggong@calliope.uwaterloo.ca).

Let r be a positive integer. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additively-written abelian groups of order r with identity $\mathcal{O}$, and let $\mathbb{G}_T$ be a multiplicatively-written cyclic group of order r with identity 1. A *bilinear pairing* on $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is a map

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$$

that satisfies the following additional properties:

1) **Bilinearity:** For $\forall P, P' \in \mathbb{G}_1$ and $\forall Q, Q' \in \mathbb{G}_2$ we have $e(P + P', Q) = e(P, Q)e(P', Q)$ and $e(P, Q + Q') = e(P, Q)e(P, Q')$.
2) **Non-degeneracy:** For $\forall P \in \mathbb{G}_1$ with $P \neq \mathcal{O}$, there is some $Q \in \mathbb{G}_2$ such that $e(P, Q) \neq 1$. Furthermore, for $\forall Q \in \mathbb{G}_2$ with $Q \neq \mathcal{O}$, there is some $P \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$.
3) **Computability:** $e(P, Q)$ can be efficiently computed for all $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$.

B. Dirichlet Multinomial Model

It is well known that the Dirichlet distribution, often denoted by $\mathsf{Dir}(\vec{\alpha})$, is a family of continuous multivariate probability distributions parameterized by the vector $\vec{\alpha}$ of positive reals which captures a sequence of observations of the possible outcomes in a state space. The Dirichlet distribution is defined as follows: Let $\Theta = \{\theta_1, \ldots, \theta_k\}$ be a state space consisting of k mutually disjoint events. Let $\vec{p} = (p(\theta_1), \ldots, p(\theta_k))$ be a continuous random vector taking values in the k-dimension simplex[1] with the joint PDF

$$f(\vec{p} \mid \vec{\alpha}) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha(\theta_i)\right)}{\prod_{i=1}^{k} \Gamma\left(\alpha(\theta_i)\right)} \prod_{i=1}^{k} p(\theta_i)^{\alpha(\theta_i)-1},$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the Gamma function. Then $\vec{p}$ is said to have a k-dimension Dirichlet distribution with parameter vector $\vec{\alpha} = (\alpha(\theta_1), \ldots, \alpha(\theta_k))$ ($\alpha(\theta_i) \geq 0$ for $i = 1, \ldots, k$). The Dirichlet distribution is the multivariate generalization of the Beta distribution. Since the Dirichlet distribution is a conjugate priori of the multinomial distribution, the posteriori distribution is also Dirichlet and can be calculated as follows [12]:

$$f(\vec{p} \mid \vec{r}, \vec{a}) = \frac{\Gamma\left(\sum_{i=1}^{k} r(\theta_i) + Ca(\theta_i)\right)}{\prod_{i=1}^{k} \Gamma\left(r(\theta_i) + Ca(\theta_i)\right)} \prod_{i=1}^{k} p(\theta_i)^{\left(r(\theta_i)+Ca(\theta_i)-1\right)}, \quad (1)$$

where $a(\theta_i)$ is a *base rate* vector over the state space Θ satisfying $a(\theta_i) \geq 0$ and $\sum_{i=1}^{k} a(\theta_i) = 1$, C is a *priori* constant which is equal to the cardinality of the state space over which a uniform distribution is assumed (C is usually set to 2), and the vector $r(\theta_i)$ is a *posteriori* evidence over the state space Θ. Given the Dirichlet distribution of Eq.(1), the probability expectation of any of the k varaiables is:

$$\mathbb{E}(p(\theta_i) \mid \vec{r}, \vec{a}) = \frac{r(\theta_i) + Ca(\theta_i)}{C + \sum_{i=1}^{k} r(\theta_i)}.$$

For more details about the Dirichlet multinomial model, the reader is referred to [8], [12].

III. System Models and Assumptions

We consider a general MANET and make the same assumptions about the network model as in [9], [22]. We term as an *adversary* or *attacker* any node whose behavior deviates from the legitimate MANET protocols. We assume that each node in a MANET is installed an Intrusion Detection System [14] which can detect predefined misbehavior. The main purpose of a key revocation scheme is to revoke keys of malicious nodes and finally isolates them from the network. Most previous schemes [13], [17], [22] are vulnerable to potentially false statement attacks in which malicious nodes provide false information about other nodes' behavior in the MANET at their own will. Therefore, we need to evaluate the influence of false statement attacks mounted by malicious nodes on our key revocation scheme in details.

We are interested in false statement attacks initiated by collusive adversaries in this paper. In this attack scenario, collusive adversaries know each other and they choose one or several well-behaving nodes as common attack objects. These adversaries always report positive observations about their friends and negative ones about the chosen victims. In this way, the adversaries can not only prolong their lifetime in the MANET, but also speed up the procedure of revoking keys of the victims. Furthermore, we also assume that adversaries always attempt to maximize their influence by propagating extremely negative observations to the network. Detailed simulations and analysis of our scheme against the above attack are presented in Section V.

IV. Protocol Description

A. Overview

Our fully self-organized key revocation scheme is within the framework of Bayesian data analysis. After an external trusted third party (TTP) bootstraps the MANET with IBC and classifies nodes' behavior into three categories, we employ Dirichlet multinomial model (see Section II-B) and explicitly use probability to quantify the uncertainty about nodes' behavior. Each node updates its global knowledge about key status of peers by observing neighbors' behavior and analyzing other nodes' reports. Moreover, IBC is used to secure the information transmission during interactions of nodes. We also add two layers of defence to thwart potential false statement attacks from malicious nodes. Firstly, a deviation test based on the statistical pattern of reports is used to filter out false statements to some extent. Furthermore, if the sender's report passes the deviation test of the receiver, we will use Dempster-Shafer belief theory to update the receiver's current knowledge about the behavior of the subject in question with this report. Based on the analysis of collected information, each node finally makes multilevel response. Our scheme consists of five parts: network initialization, neighborhood watch, authenticated information dissemination, filter of false statements, and multilevel response for malicious nodes. A high level description of our key revocation scheme is shown in the following Algorithm 1.

[1] The k-dimension simplex is such that if $\vec{p} = (p(\theta_1), \ldots, p(\theta_k))$ then $p(\theta_i) \geq 0$ and $\sum_{i=1}^{k} p(\theta_i) = 1$.

```
Algorithm 1 Self-Organized Key Revocation for MANETs
Step 1. Network Initialization
        ▷ Generation of system parameters
        ▷ Registration of network nodes
        ▷ Classification of node behavior
Step 2. Neighborhood Watch
        ▷ Monitor neighbors' behavior and generate observation matrix
        ▷ Update key status of nodes with direct observations
Step 3. Authenticated Information Dissemination
        ▷ Disseminate nodes' direct observations to all $m$-hop neighbors
          in an authenticated way by using a keyed-hash function
Step 4. Filter of False Statements
        ▷ Filter out potentially false statements statistically
        ▷ Update key status of nodes based on Dempster-Shafer theory
Step 5. Multilevel Response for Malicious Nodes
        ▷ Revoke keys of nodes showing malicious behavior
        ▷ Cease communication with nodes showing suspicious behavior
          and keep observing their behavior for further decision
```

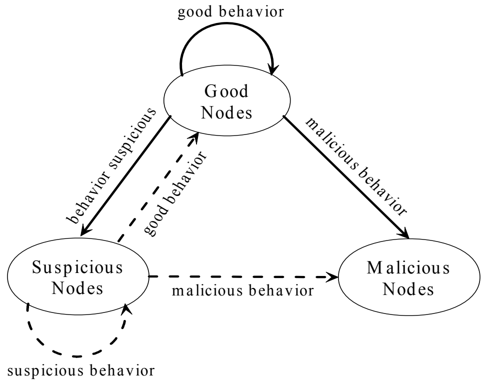


Fig. 1. State Transition Diagram among Different Types of Nodes

B. Step 1. Network Initialization

Our scheme assumes that an external TTP bootstraps the MANET with IBC and classifies the behavior of nodes. More specifically, the external TTP will complete the following tasks during network initialization:

1) Generation of system parameters: The TTP generates secure system parameters $\langle q, k, C/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e\rangle$ as described in Section II-A. Note that we take $\mathbb{G}_1 = \mathbb{G}_2$ in this paper. The TTP also generates a random master key $s \in \mathbb{Z}_n^*$ and a random generator $P \in \mathbb{G}_1$, and sets his public key $P_{pub} = sP \in \mathbb{G}_1$. Finally, the TTP chooses a cryptographic secure hash function: $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. The TTP publishes all of these parameters except his master key.

2) Registration of network nodes: For the purpose of key revocation, we use the public key format $Q_i = H(ID_i \parallel \mathsf{date} \parallel \mathsf{version})$ for each node with identity ID_i as introduced in [9], where date is the expiry date of the key and $\mathsf{version}$ is its version number. The TTP also generates the ID-based private key $d_i = sQ_i$ for node ID_i.

3) Classification of node behavior: In our model, the state space Θ includes three mutually disjoint events: good behavior θ_g, suspicious behavior θ_s and malicious behavior θ_m, namely $\Theta = \{\theta_g, \theta_s, \theta_m\}$. To keep track of various observable behavior in the lifetime of the MANET, the TTP classifies nodes' behavior into three categories, namely good behavior set $\mathbb{B}_g$, suspicious behavior set $\mathbb{B}_s$ and malicious behavior set $\mathbb{B}_m$. The set $\mathbb{B}_g$ includes behavior complying with descriptions of the MANET protocols. The set $\mathbb{B}_s$ contains accidental misbehavior that temporarily and slightly deteriorate the performance of MANETs, whereas intentional misbehavior, which seriously degrade the performance of MANETs, are comprised in the set $\mathbb{B}_m$. The practical classification of the sets $\mathbb{B}_g, \mathbb{B}_s$ and $\mathbb{B}_m$ depends on the network policy, the detection ability of nodes and the concrete application scenarios. Fig. 1 demonstrates possible state transitions among different types of nodes in the lifetime of the MANET. Since nodes' behavior must fall into one of the above three categories, nodes analyze and predict peers' behavior with 3-dimension Dirichlet distribution $\mathsf{Dir}(\alpha_g, \alpha_s, \alpha_m)$ (see Section II-B), where $(\alpha_g, \alpha_s, \alpha_m)$ is a parameter vector which keeps track of nodes' behavior appearing in sets $\mathbb{B}_g, \mathbb{B}_s$ and $\mathbb{B}_m$, respectively.

After the network initialization phase, each node ID_i is preloaded the following materials:

- **System Parameters**: $\langle q, k, C/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_T, e, H, P, P_{pub}\rangle$.
- **Public / Private Key Pair**: $\langle Q_i, d_i\rangle$.
- **Behavior Classification:** $\mathbb{B}_g, \mathbb{B}_s$ and $\mathbb{B}_m$.

C. Step 2. Neighborhood Watch

In the neighborhood watch scheme, each node ID_i monitors all its one-hop neighbors and records three categories of behavior each time they occur. We do not limit types of node behavior and any new type of observable behavior can be added to the corresponding set $\mathbb{B}_g, \mathbb{B}_s$ or $\mathbb{B}_m$.

Let $\mathcal{N}_i^{(1)}$ be the set of one-hop neighbors of node ID_i. We use the parameter vector $\left(\gamma^i_{j,g}, \gamma^i_{j,s}, \gamma^i_{j,m}\right)$ of 3-dimension Dirichlet distribution to record node ID_i's direct experience with the node ID_j. Initially, the parameter vector is set to $(Ca(\theta_g), Ca(\theta_s), Ca(\theta_m))$, where $(a(\theta_g), a(\theta_s), a(\theta_m))$ is the base rate vector and C is the prior constant (see Section II-B). Node ID_i makes one individual observation for each node $ID_j \in \mathcal{N}_i^{(1)}$ periodically. We set binary variables $\beta^i_{j,g}, \beta^i_{j,s}$ and $\beta^i_{j,m}$ to be 1 if node ID_i's observation about node ID_j's behavior is classified into the sets $\mathbb{B}_g, \mathbb{B}_s$ or $\mathbb{B}_m$, respectively, and 0 otherwise. According to new observations about behavior of all its one-hop neighbors, node ID_i first updates its direct experience for each $ID_j \in \mathcal{N}_i^{(1)}$ with the following formulae:

$$\gamma^i_{j,x} = \mu\gamma^i_{j,x} + \beta^i_{j,x}, x \in \{g, s, m\},$$

where the weight $\mu \in [0, 1]$ is a discount factor for past observations (typically, μ is very close to 1). Node ID_i then updates its own *observation matrix* OM^i with new information. Assume that node ID_i has obtained direct experience with N_i nodes in the network up to the current time instance, node ID_i's observation matrix is as follows:

$$OM^i = \begin{bmatrix} ID_1 & \gamma^i_{1,g} & \gamma^i_{1,s} & \gamma^i_{1,m} \\ \vdots & \vdots & \vdots & \vdots \\ ID_{N_i} & \gamma^i_{N_i,g} & \gamma^i_{N_i,s} & \gamma^i_{N_i,m} \end{bmatrix}.$$

We use the parameter vector $(\alpha^i_{j,g}, \alpha^i_{j,s}, \alpha^i_{j,m})$ of 3-dimension Dirichlet distribution to keep track of node ID_i's accumulated knowledge about node ID_j's behavior. Note that the vector $(\alpha^i_{j,g}, \alpha^i_{j,s},\ \alpha^i_{j,m})$ will be updated by both node

ID_i's direct experience and reports from other nodes. Initially, the parameter vector is also set to $(Ca(\theta_g), Ca(\theta_s), Ca(\theta_m))$. After node ID_i makes a direct observation about node ID_j's behavior, its global knowledge about node ID_j's behavior will be updated with the following formulae:

$$\alpha^i_{j,x} = \mu\alpha^i_{j,x} + \beta^i_{j,x}, x \in \{g, s, m\}.$$

Upon obtaining new information about all its one-hop neighbors, node ID_i also updates corresponding rows in its *node status matrix*, NSM^i, which indicates node ID_i's opinion about key status of other nodes. Let N be the total number of nodes in the MANET. Furthermore, we assume that node ID_i has obtained the knowledge of key status of M_i nodes until the current time instance by observing its one-hop neighbors and collecting information from others. Then node ID_i's *node status matrix* NSM^i is as follows:

$$NSM^i = \begin{bmatrix} ID_1 & (t^i_1, v^i_1) & R^i_1 & \alpha^i_{1,g} & \alpha^i_{1,s} & \alpha^i_{1,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_{M_i} & (t^i_{M_i}, v^i_{M_i}) & R^i_{M_i} & \alpha^i_{M_i,g} & \alpha^i_{M_i,s} & \alpha^i_{M_i,m} \\ ID_{M_i+1} & ? & ? & Ca(\theta_g) & Ca(\theta_s) & Ca(\theta_m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_N & ? & ? & Ca(\theta_g) & Ca(\theta_s) & Ca(\theta_m) \end{bmatrix},$$

where t^i_j and v^i_j represent the expiry date and version number of node ID_j's public key, respectively. $R^i_j \in \{-1, 0, 1\}$ denotes key status of node ID_j from the point of view of node ID_i, and "?" means node ID_i does not obtain any information about behavior of nodes $ID_k, k \in \{M_i + 1, \ldots, N\}$ until the current time instance. Note that R^i_j being -1, 0 or 1 indicates that the status of node ID_j's key is "Revoked", "Suspicious" or "Trustworthy", respectively.

D. Step 3. *Authenticated Information Dissemination*

Periodically, node ID_i securely disseminates its direct experience about other nodes' behavior to all its m-hop neighbors. Let $\mathcal{N}^{(m)}_i$ be the set of m-hop neighbors of node ID_i. Node ID_i then sends its observation matrix OM^i to each node $ID_j \in \mathcal{N}^{(m)}_i$ with the following format:

$$om^i_j = ((ID_i, ID_j, OM^i), h_{K_{i,j}}(ID_i, ID_j, OM^i)),$$

where $K_{i,j}$ is the pre-shared key between a pair of nodes ID_i and ID_j, and $h_{K_{i,j}}(\cdot)$ is a secure hash function taking $K_{i,j}$ as the input key. With the aid of the cryptographic pairing (see Section II-A), the pre-shared key $K_{i,j}$ can be separately calculated by nodes ID_i and ID_j in a non-interactive fashion during the phase of a neighbor discovery as follows:

$$K_{i,j} = e(d_i, Q_j) = e(sQ_i, Q_j) = e(Q_i, sQ_j) = e(Q_i, d_j),$$

where $\langle Q_i, d_i\rangle$ and $\langle Q_j, d_j\rangle$ are the public/private key pair of nodes ID_i and ID_j, respectively. Furthermore, both data integrity and authenticity of messages are guaranteed by the keyed-hash function $h_{K_{i,j}}(\cdot)$. Therefore, an attacker cannot change content of the observation matrix.

E. Step 4. *Filter of False Statements*

Each time node ID_i receives an observation matrix om^j_i from node ID_j, node ID_i will perform the following information processing and integration algorithm shown in Fig. 2.

In Step 4.1, node ID_i checks the status of node ID_j's key in the node status matrix NSM^i. If $R^i_j = 1$, then node ID_i considers node ID_j to be trustworthy and continues the next step; otherwise node ID_i will discard the observation matrix received from node ID_j and stop.

In Step 4.2, node ID_i verifies the authenticity of the message om^j_i using the pre-shared key $K_{i,j}$, as described in Section IV-D. If the message passes the authentication, node ID_i will further analyze reliability of node ID_j's observation in Step 4.3, otherwise node ID_i knows that the received message does not come from node ID_j, and therefore just discards it and stops.

Due to the possibility that nodes are compromised and then arbitrarily report their observations under the control of attackers, messages that node ID_i receives from its counterparts might be spurious. Therefore, the main purpose of Step 4.3 is to avoid or mitigate the influence of false statements from malicious nodes to some degree. In the context of key revocation, attackers' goals are twofold by manipulating observations of compromised nodes. On the one hand, attackers can choose one or many good nodes and report unfairly negative observations about victims' behavior in order to revoke their keys. On the other hand, if attackers know each other and collude in MANETs, they will also propagate unfairly positive observations about their confederates' behavior for the purpose of keeping their keys valid and further damaging the operation of the network. Two efficient statistical filtering techniques based on Beta distribution have been proposed to protect Bayesian reputation systems from liars by Whitby *et al.* [20] and Buchegger *et al.* [5], [6], respectively. Their methods are based on the assumption that the statistical pattern of dishonest reports is different from that of truthful ones. In addition, the difference between these two techniques is that Whitby *et al.*'s method uses quantiles of Beta distribution, whereas Buchegger *et al.*'s method employs a deviation test for the compatibility of received messages. Since our key revocation scheme is based

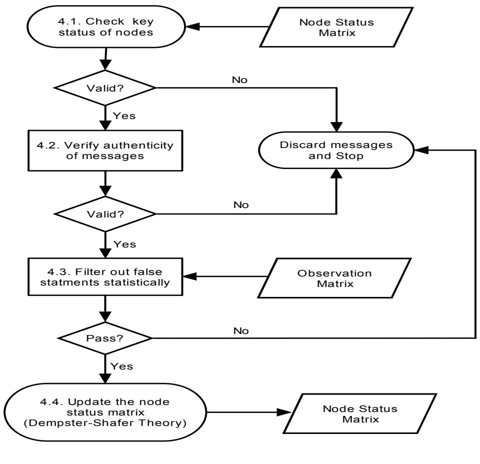

4.1. Check key status of nodes
Node Status Matrix
Valid?
No
Yes
4.2. Verify authenticity of messages
Discard messages and Stop
Valid?
No
Yes
4.3. Filter out false statments statistically
Observation Matrix
Pass?
No
Yes
4.4. Update the node status matrix (Dempster-Shafer Theory)
Node Status Matrix

Fig. 2. Information Processing and Integration Algorithm

on the Dirichlet distribution and it is difficult to define the quantile in the multivariate case, we only generalize the idea of the deviation test suggested by Buchegger *et al.* [5], [6] to Dirichlet multinomial model in this work.

In Step 4.3, node ID_i extracts orderly each row from the node ID_j's observation matrix OM^j and performs a deviation test for the compatibility of node ID_j's observations. More specifically, when node ID_i extracts the k-th row from OM^j, it computes the following two posteriori expected probabilities with which node ID_k shows behavior in $\mathbb{B}_s$ and $\mathbb{B}_m$, respectively:

$$\begin{aligned}\mathbb{E}\left(p(\theta_s) \mid \vec{\gamma}^j_k, \vec{a}\right) &= \frac{\gamma^j_{k,s} + Ca(\theta_s)}{C + \gamma^j_{k,g} + \gamma^j_{k,s} + \gamma^j_{k,m}},\\ \mathbb{E}\left(p(\theta_m) \mid \vec{\gamma}^j_k, \vec{a}\right) &= \frac{\gamma^j_{k,m} + Ca(\theta_m)}{C + \gamma^j_{k,g} + \gamma^j_{k,s} + \gamma^j_{k,m}},\end{aligned}$$

where $\vec{\gamma}^j_k = \left(\gamma^j_{k,g}, \gamma^j_{k,s}, \gamma^j_{k,m}\right)$ represents node ID_j's direct experience about node ID_k's behavior, and $\vec{a} = (a(\theta_g), a(\theta_s), a(\theta_m))$ is the default base rate vector. And then, node ID_i takes the row corresponding to node ID_k from its node status matrix NSM^i and separately calculates two expected probabilities based on its own knowledge about node ID_k's behavior as follows:

$$\begin{aligned}\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}^i_k, \vec{a}\right) &= \frac{\alpha^i_{k,s} + Ca(\theta_s)}{C + \alpha^i_{k,g} + \alpha^i_{k,s} + \alpha^i_{k,m}},\\ \mathbb{E}\left(p(\theta_m) \mid \vec{\alpha}^i_k, \vec{a}\right) &= \frac{\alpha^i_{k,m} + Ca(\theta_m)}{C + \alpha^i_{k,g} + \alpha^i_{k,s} + \alpha^i_{k,m}},\end{aligned}$$

where $\vec{\alpha}^i_k = \left(\alpha^i_{k,g}, \alpha^i_{k,s}, \alpha^i_{k,m}\right)$ denotes node ID_i's accumulated knowledge about node ID_k's behavior. After obtaining the above four expected probabilities, node ID_i executes the following deviation tests:

$$\begin{aligned}\left|\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}^i_k, \vec{a}\right) - \mathbb{E}\left(p(\theta_s) \mid \vec{\gamma}^j_k, \vec{a}\right)\right| &\leq \varepsilon_1,\\ \left|\mathbb{E}\left(p(\theta_m) \mid \vec{\alpha}^i_k, \vec{a}\right) - \mathbb{E}\left(p(\theta_m) \mid \vec{\gamma}^j_k, \vec{a}\right)\right| &\leq \varepsilon_2,\end{aligned}$$

where $\varepsilon_1, \varepsilon_2 \in (0,1)$ are two deviation thresholds determined by a system designer. If node ID_j's report about node ID_k's behavior cannot pass the above deviation tests, node ID_i considers that report as incompatible and just discards it. Otherwise, node ID_i uses node ID_j's report to update its knowledge about the behavior of the node ID_k in Step 4.4.

Note that the simplistic information integration method used in [6] is vulnerable to false statement attacks from an adversary, as analyzed theoretically in [16]. Therefore, we set up the second defense line to thwart false statement attacks by integrating other nodes' reports based on Dampster-Shafer belief theory [18]. In [10], Jøsang constructed a bijective mapping between Dirichlet distributions and Dampster-Shafer belief functions. Therefore, we first map node ID_i's accumulated knowledge and node ID_j's report about node ID_k's behavior (two Dirichlet distributions) to two belief distribution functions, respectively. Then we use the technique of belief discounting [11] to update node ID_i's opinion about node ID_k's behavior as a result of node ID_j's report. Finally we map the resulting belief function to a Dirichlet distribution. In this way, the reports from different nodes are given different weight based on their respective reputation. Suppose that

$$\nu = \frac{C\alpha^i_{j,g}}{\left(C + \alpha^i_{j,s} + \alpha^i_{j,m}\right)\left(C + \gamma^j_{k,g} + \gamma^j_{k,s} + \gamma^j_{k,m}\right) + C\alpha^i_{j,g}}.$$

Then node ID_i uses node ID_j's report to update its global knowledge about node ID_k's behavior with the following equations:

$$\alpha^i_{k,x} = \alpha^i_{k,x} + \nu\gamma^j_{k,x}, x \in \{g, s, m\}.$$

F. Step 5. *Multilevel Response for Malicious Nodes*

Each time node ID_i updates its knowledge about node ID_k's behavior in the MANET by either the neighborhood watch scheme or other nodes' reports, it checks whether ID_k's behavior are still within boundaries of its misbehavior tolerance and the status of node ID_k's key needs to be changed. Note that node ID_k's key status R^i_k in the node status matrix NSM^i directly determines how node ID_i treats node ID_k.

To minimize the squared-error loss for the deviation from the true probabilities $p(\theta_m)$ and $p(\theta_s)$ with which node ID_k shows respectively malicious and suspicious behavior, we choose posteriori expected probabilities $\mathbb{E}\left(p(\theta_m) \mid \vec{\alpha}^i_k, \vec{a}\right)$ and $\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}^i_k, \vec{a}\right)$ as estimators as usually done. As soon as node ID_i obtains the updated vector $\vec{\alpha}^i_k$ describing node ID_k's behavior, it will response as follows:

1. Node ID_i computes the posteriori expected probability $\mathbb{E}\left(p(\theta_m) \mid \vec{\alpha}^i_k, \vec{a}\right)$. If $\mathbb{E}\left(p(\theta_m) \mid \vec{\alpha}^i_k, \vec{a}\right) \geq t_{rev}$, i.e., it is equal to or larger than a predetermined revocation threshold t_{rev}, node ID_i sets $R^i_k = -1$ and stops. Otherwise it goes to the next step. Here, $R^i_k = -1$ denotes that node ID_i believes that node ID_k has been compromised and revokes its key. Once node ID_i revokes node ID_k's key, it will cease any communication with node ID_k until node ID_k receives a new key from the TTP.
2. Node ID_i calculates the posteriori expected probability $\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}^i_k, \vec{a}\right)$. If $\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}^i_k, \vec{a}\right) \geq t^k_{sus}$, i.e., it is equal to or larger than a predetermined suspicion threshold t^k_{sus}, node ID_i sets $R^i_k = 0$. Note that $R^i_k = 0$ means that node ID_i suspects that node ID_k has been compromised, and so node ID_i will shield itself against suspicious behavior of node ID_k by terminating the communication with it. Furthermore, to make further decision, node ID_i continues collecting information to update its knowledge about node ID_k's behavior. Possible state transitions of the suspicious node ID_k are described with dash lines in Fig. 1. Note that three cases might happen for node ID_k: a) Node ID_k just shows suspicious behavior by accident, and therefore behaves normally after a short time. In this case, it will become a good node and be trusted by node ID_i again. b) Node ID_k continues behaving suspiciously. In this case, all nodes will finally mark node ID_k to be suspicious and terminate to communicate with it. Hence, node ID_k will be evicted from the network. c) Node ID_k shows

malicious behavior. In this case, the key of node ID_k will be revoked once the posterior expected probability $\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}_k^i, \vec{a}\right)$ reaches the revocation threshold. In addition, to react faster than before when node ID_k behaves suspiciously again in the above case a), node ID_i also decreases the suspicion threshold of node ID_k as follows:

$$t_{sus}^k := \xi t_{sus}^k,$$

where $\xi \in (0,1)$ is a fading factor of the suspicion threshold of a node. Furthermore, we also introduce a parameter t_{max} which denotes the maximum number of state transitions between good nodes and suspicious nodes (see Fig. 1). Once the state transition has appeared t_{max} times for node ID_k, node ID_i will revoke its key immediately by setting $R_k^i = -1$ and terminate any further communication with node ID_k until node ID_k receives a new key from the TTP.

V. Performance Evaluation

In this section, we evaluate the performance of our key revocation scheme through extensive simulations, the goal of which is to demonstrate attack-resistant properties of our scheme under the existence of collusive adversaries. Furthermore, we also show the advantages of classifying nodes' behavior into three categories over the simple binary differentiation.

A. Simulation Setup

we have implemented our key revocation scheme with the C programming language on *Microsoft Visual Studio* platform. The performance evaluations are based on the simulations of 100 wireless nodes that form a MANET over a square (600m×600m) space and interact 100 times. We use the "random waypoint" model [3] to simulate the mobility of nodes in the MANET. For each node, we set the speed to be uniformly distributed between 5 and 20 m/s. The minimum speed is chosen so high to exclude the case of the static network [21]. The communication range of each node is set to be 150 m. Furthermore, we assume that the base rate vector $\vec{a} = (a(\theta_g), a(\theta_s), a(\theta_m))$ is $(0.6, 0.25, 0.15)$, which denotes the prior uncertainty that honest nodes show good behavior, suspicious behavior and malicious behavior, respectively. We also assume that the discount factor μ is 0.999, both deviation thresholds ε_1 and ε_2 are 0.1, the revocation threshold t_{rev} is 0.2 and the suspicious threshold t_{sus}^k is 0.3. The simulation is repeated for a number of communication sessions. In each session, each node moves to a new position and observe the behavior of its neighbors. Moreover, in some sessions nodes also flood their observations to all one-hop neighbors.

To simulate false statement attacks from adversaries, before running the simulation, we randomly select a certain fraction of the network population as suspicious nodes and malicious nodes, respectively. More specifically, we assume that 20% of all network nodes will show suspicious behavior for different reasons. Among those suspicious nodes, we further assume that half of them, named *type-I suspicious nodes*, show suspicious behavior just by accident (for example, a node drops packages due to the network congestion.) and behave normally after some time (due to the improvements of the network environment), whereas the other half of suspicious nodes, called *type-II suspicious nodes*, show suspicious behavior followed by malicious behavior. Note that type-I suspicious nodes are basically good and therefore record their observations honestly, whereas type-II suspicious nodes are basically malicious and so we assume that they record a malicious behavior for selected attack objects in each communication session. Considering two types of suspicious nodes in the simulations enables us to demonstrate the following two cases (also see Fig. 1): a) Type-I suspicious nodes can get trustworthy again by good nodes after they are marked as suspicious; b) Keys of type-II suspicious nodes will be finally revoked. Furthermore, we also change the fraction of malicious nodes, ranging from 10% to 30%. Based on the above parameters and assumptions, we simulate the attack scenario described in Section III.

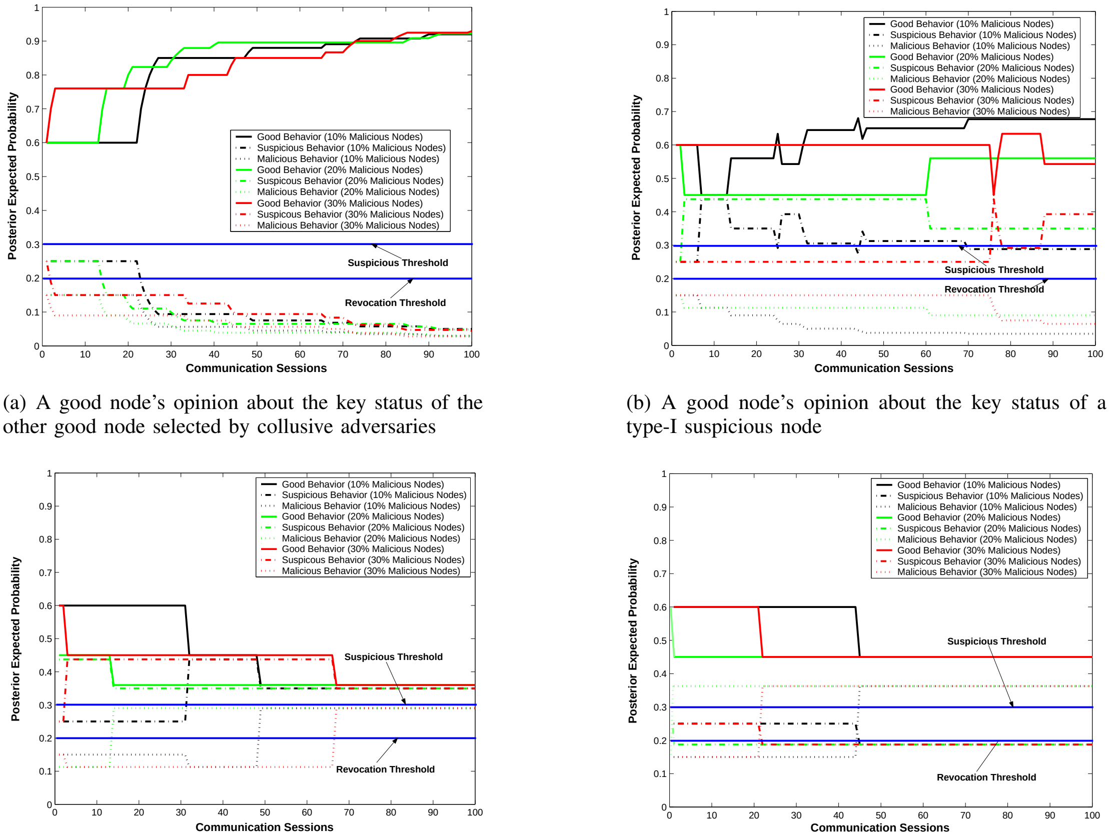
B. False Statement Attacks by Collusive Adversaries

In this section, we study whether false statement attacks from collusive adversaries will affect our key revocation scheme. To this end, we assume that all malicious nodes choose 10% good nodes as common targets instead of randomly and independently selecting attack objects. In this attack scenario, all malicious nodes not only record malicious behavior for the selected 10% good nodes but also record good behavior for other malicious nodes in each communication session. Furthermore, they also propagate their false statements to all one-hop neighbors each 5 communication sessions.

Note that we are concerned with the opinion of a good node about the key status of other nodes under the false statement attack by collusive adversaries. Therefore, we randomly select two good nodes (one of them is the attack object of the collusive adversaries), a type-I suspicious node, a type-II suspicious node, and a malicious node, and keep track of the opinion of a good node. Figure 3 shows the attack-resistance properties of our key revocation scheme against collusive adversaries when the number of malicious nodes increases from 10% to 30%. We want to emphasize again that in our key revocation scheme each node has its own view about the key status of other nodes. Although we observe that all good nodes have similar opinion about other nodes' key status in our simulations, it is impossible for us to show all good nodes' opinion due to space limitations. Therefore, we randomly sample several nodes from different categories.

In Figure 3(a), we note that false accusations from collusive adversaries cannot affect the good node's opinion about the key status of the victim they select. The posterior expected probability that the victim shows malicious behavior is always less than the revocation threshold. The reason is that good nodes have accumulated good reputation in the early communication sessions and the false accusations from adversaries cannot pass the deviation test set by good nodes. Therefore, the false accusations will be filtered by good nodes and the keys of good nodes will not be wrongly revoked even in the presence of collusive adversaries.

Figure 3(b) shows that the key of a type-I suspicious node will not be revoked by the good node unless the number of



(a) A good node's opinion about the key status of the other good node selected by collusive adversaries

(b) A good node's opinion about the key status of a type-I suspicious node

(c) A good node's opinion about the key status of a type-II suspicious node

(d) A good node's opinion about the key status of a malicious node

Fig. 3. Simulation Results for False Statement Attacks by Collusive Adversaries

times that it changes its states between good and suspicious amount to t_{max} (see Fig. 1). Furthermore, if the key of the type-I suspicious node is marked as suspicious due to temporary suspicious behavior, it can be trusted again by a good node after the posterior expected probability $\mathbb{E}\left(p(\theta_s) \mid \vec{\alpha}_k^{i,new}, \vec{a}\right)$ is less than the suspicious threshold. Different from type-I suspicious nodes, malicious behavior of a type-II suspicious node are finally identified by the good node and therefore it will revoke the key of the type-II suspicious node as shown in Figure 3(c). Figure 3(b) and 3(c) demonstrate how a good node responses suspicious behavior in our scheme. While a good node showing suspicious behavior temporarily can get trustworthy again by other good nodes, the real malicious nodes will be evicted from the network. Moreover, false statement attacks from collusive adversaries have no influence on type-I and type-II suspicious nodes since the attack objects of adversaries are good nodes in our simulations.

Figure 3(d) shows that the key of the malicious node can still be revoked by the good node even if malicious nodes praise each other. The reason is that after the good node have established the bad reputation for the malicious node in the early communication sessions the false praise from the friends of the malicious node is very difficult to pass the deviation test of good nodes. Moreover, even if the false statement can pass the deviation test, this information only has slight influence on the opinion of the good node because of the use of Dampster-Shafter theory (see Section IV-E), which gives less weight to the reports from malicious nodes than those from good nodes.

The simulation results in Figure 3 demonstrate that false statements from collusive malicious nodes cannot affect good nodes' opinion about the key status of other nodes. Most false statements are filtered by the deviation tests of good nodes. For those false statements which pass the deviation tests, the information integration technique based on Dampster-Shafter theory guarantees that the false statements only have slight influence on good nodes' opinion. Therefore, our key revocation can still perform well even under the false statement attacks from collusive adversaries.

VI. Related Work

In this section, we briefly review previous key revocation schemes and reputation systems related to our work.

Luo *et al.* [13] presented a certificate revocation and key update scheme based on CBC and threshold cryptography. Similar idea is also implemented with IBC in [17]. In these schemes, each node monitors the behavior of its one-hop neighboring nodes and disseminates its signed accusations to its m-hop neighborhoods upon observing any malicious behavior, where m is a design parameter denoting the range of the accusation propagation. All nodes receiving the accusation information verify whether the accuser can be trusted and update their CRLs accordingly. When the number of accusations for some node exceeds a predefined revocation threshold, the certificate of that node will be revoked. Later on, Zhang *et al.* [22] proposed an threshold signature scheme to revoke keys of malicious nodes by combining IBC and threshold cryptography. Although these revocation schemes do not require the set-up of any infrastructure, the threshold scheme introduces a lot of communication and computational overhead to the network. Furthermore, all the schemes above are vulnerable to the false accusation attacks and malicious nodes can collude each other to revoke keys of good nodes.

Several fully self-organized key revocation schemes are also proposed in the literature. Moore *et al.* [15] introduced the concept of suicide for solving the credential revocation in self-organizing systems. When a node observes the misbehavior of another node, it simply broadcasts a signed message claiming both of them to be dead. Their scheme can fast isolate the malicious nodes from the network at the cost of evicting the same amount of good nodes either. In [2], Arboit *et al.* proposed a certificate revocation protocol using a weighted accusation scheme. However, All accusations are frequently broadcasted throughout the entire network in their scheme and therefore the communication cost is very high. Hoeper and Gong [9] also presented a self-organized key revocation scheme in which the nodes' observations are then securely propagated to m-hop neighborhoods using preshared keys obtained from a non-interactive ID-based key agreement protocol. Furthermore, the majority vote is used to mitigate the influence of false accusations from malicious nodes.

All key revocation schemes above only classify the node as either good or bad without any intermediate state. Therefore, good nodes who only misbehave for a short time due to various reasons (for example, the network congestion) might be evicted from the network. Compared to previous schemes, our scheme does not immediately revoke keys of suspicious nodes and keeps collecting information for making more precise estimation about nodes' behavior once the nodes are marked as suspicious. Furthermore, our scheme also borrows the idea from the reputation system CONFIDANT [5], [7] to mitigate the influence of potentially false statement attacks from malicious nodes.

VII. Conclusion

MANETs pose formidable challenges on the issue of key revocation due to lack of infrastructure and centralized servers. This work explores a novel self-organized approach to solve the key revocation problem in MANETs. Firmly rooted in statistics, our key revocation scheme provides a theoretically sound basis for nodes analyzing and predicting peers' behavior based on their own observations and other nodes' reports. Furthermore, classifying nodes' behavior into three categories not only provides network designers more flexibility for various application scenarios, but also enables nodes to make multi-level response according to the severity of malicious behavior. In addition, our key revocation scheme is designed to provide strong defense against false statement attacks from collusive adversaries. The effectiveness and attack-resistance properties of our scheme are confirmed by extensive simulation results.

References

[1] F. Anjum, and P. Mouchtaris, *Security for Wireless Ad Hoc Networks*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[2] G. Arboit, C. Crépeau, C. R. Davis, and M. Maheswaran, "A Localized Certificate Revocation Scheme for Mobile Ad Hoc Networks," *Ad Hoc Network*, vol.6, no.1, pp. 17-31, 2008.

[3] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pp. 85-97, 1998.

[4] L. Buttyan, and J.-P. Hubaux, *Security and Cooperation in Wireless Networks*, Cambridge University Press, 2007.

[5] S. Buchegger, and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes – Fairness In Dynamic Ad-hoc Networks," *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pp. 226-236, 2002.

[6] S. Buchegger, and J.-Y. Le Boudec, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks," *Proceedings of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[7] S. Buchegger, and J.-Y. Le Boudec, "A Robust Reputation System for Peer-to-Peer and Mobile Ad Hoc Networks," *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[8] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis, Second Edition*, Boca Raton, Florida, USA: Chapman & Hall/CRC, 2004.

[9] K. Hoeper, and G. Gong, "Key Revocation for Identity-Based Schemes in Mobile Ad Hoc Networks," *Proceedings of the 5th International Conference on Ad-Hoc, Mobile, and Wireless Networks - ADHOC-NOW 2006*, ser. LNCS 4104, pp. 224-237, 2006.

[10] A. Jøsang, "Probabilistic Logic Under Uncertainty," *Proceedings of the thirteenth Australasian symposium on Theory of computing - Volume 65*, pp. 101-110, 2007.

[11] A. Jøsang, and R. Ismail, "The Beta Reputation Systems," *Proceedings of the 15th Bled Electronic Commerce Conference - eReality: Constructing the eEconomy*, pp. 324-337, 2002.

[12] A. Jøsang, and J. Haller, "Dirichlet Reputation Systems," *Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES 2007)*, pp. 112-119, 2007.

[13] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Roubust Access Control for Mobile Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, vol.12, no.6, pp. 1049-1063, 2004.

[14] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks," *IEEE Wireless Communication*, vol. 11, no. 1, pp. 48-60, Feb. 2004.

[15] T. Moore, J. Clulow, R. Anderson, and S. Nagaraja, "New Strategies for Revocation in Ad Hoc Networks," *Proceedings of the Fourth European Workshop on Security and Privacy in Ad Hoc and Sensor Networks - ESAS 2007*, ser. LNCS 4572, pp. 232-246, 2007.

[16] J. Mundinger, and J.-Y. Le Boudec, "Analysis of A Reputation System for Mobile Ad-hoc Networks with Liars," *Proceedings of WiOpt 2005: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 41-46, 2005.

[17] N. Saxena, G. Tsudik, and J. H. Yi, "Identity-Based Access Control for Ad Hoc Groups," *Proceedings of the 7th International Conference on Information Security and Cryptology - ICISC 2004*, ser. LNCS 3506, pp. 362-379, 2004.

[18] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University, 1976.

[19] A. Shamir, "Identity Based Cryptosystems and Signature Schemes," *Proceedings of Advances in Cryptology - CRYPTO 1984*, ser. LNCS 196, pp. 47-53, 1984.

[20] A. Withby, A. Jøsang, and J. Indulska, "Filtering Out Unfair Ratings in Bayesian Reputation Systems," *The Icfain Journal of Management Research*, vol. 4, no. 2, pp. 48-64, 2005.

[21] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM 2003*, pp. 1312-1321, 2003.

[22] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing Mobile Ad Hoc Networks with Certificateless Public Keys," *IEEE Transactions on Dependable and Secure Computing*, vol.3, no. 4, pp. 386-399, OCTOBER-DECEMBER 2006.