Securing Cooperative Ad-Hoc Networks Under Noise and Imperfect Monitoring: Strategies and Game Theoretic Analysis

Wei Yu, Zhu Ji, and K. J. Ray Liu, *Fellow, IEEE*

Abstract—**In cooperative ad-hoc networks, nodes belong to the same authority and pursue the common goals, and will usually unconditionally help each other. Consequently, without necessary countermeasures, such networks are extremely vulnerable to insider attacks, especially under noise and imperfect monitoring. In this paper, we present a game theoretic analysis of securing cooperative ad-hoc networks against insider attacks in the presence of noise and imperfect monitoring. By focusing on the most basic networking function, namely routing and packet forwarding, we model the interactions between good nodes and insider attackers as secure routing and packet forwarding games. The worst case scenarios are studied where initially good nodes do not know who the attackers are while insider attackers know who are good. The optimal defense strategies have been devised in the sense that no other strategies can further increase the good nodes' payoff under attacks. Meanwhile, the optimal attacking strategies and the maximum possible damage that can be caused by attackers have been discussed. Extensive simulation studies have also been conducted to evaluate the effectiveness of the proposed strategies.**

Index Terms—**Ad-hoc networks, game theory, security.**

I. Introduction

An ad-hoc network is a group of nodes without requiring a fixed network infrastructure, where nodes can communicate with others out of their direct transmission ranges by cooperatively forwarding packets for each other. In many applications, such as military or emergency situations, nodes in an ad-hoc network belong to the same authority and pursue a common goal. Therefore, fully cooperative behavior, such as unconditionally forwarding packets for each other, can usually be assumed. We refer to such ad-hoc networks as cooperative ad-hoc networks.

Before ad-hoc networks can be successfully deployed, however, security issues must be resolved first [1]–[9]. Past experiences have shown that security in ad-hoc networks is particularly hard to achieve: nodes lacking enough physical protection can be easily captured, compromised, and hijacked; the dynamically changing topology and fragile wireless links may result in high link breakage ratio. Moreover, for cooperative ad-hoc networks, the fully cooperative nature makes them extremely vulnerable to insider attacks. In the literature, many schemes have been proposed to prevent attackers from entering the network through secure key distribution and secure neighbor discovery [1], [4]–[6], [10]–[14]. But these schemes are not effective in the presence of insider attackers, that is, when attackers have gained access to the network.

In the literature, several schemes have been proposed to secure ad-hoc networks against insider attacks based on necessary monitoring [3], [9], [15]–[18]. In [3], the "watchdog" mechanism was proposed to mitigate nodes' misbehavior in ad-hoc networks. Following that, CONFIDANT was proposed to detect and isolate misbehaving nodes [15], and CORE was proposed to enforce cooperation among selfish nodes [16]. Recently, HADOF was proposed to defend against routing disruption attacks [8], and an effective header watcher mechanism was proposed to defend against injecting traffic attacks [9], both targeting cooperative ad-hoc networks.

However, there still exist some important issues which have not been fully addressed. One is the optimality measure of defense mechanisms. For example, what metrics should be used to measure the optimality of the defense mechanism? Under certain optimality metrics, what are the optimal defending strategies, especially when the environment is noisy and the monitoring is not perfect? What strategies should the attackers use to maximize the damage to the network and, consequently, what is the maximum possible damage that the attackers can cause? No existing work has fully addressed these issues.

In this paper, we jointly consider routing and packet forwarding in cooperative ad-hoc networks, and model the interactions between good nodes and attackers as games, referred to as secure routing and packet forwarding games. We adopt Nash equilibrium[1] as a basic optimality metric. In order to fully address the above issues, we focus on the following scenario: initially good nodes do not know who attackers are while attackers can know who the good nodes are. This scenario

Manuscript received February 21, 2006; revised February 23, 2007. This work was supported in part by the Army Research Office under URI Award No. DAAD19-01-1-0494. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Klara Nahrstedt.

W. Yu was with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: weiyu@isr.umd.edu). He is now with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: weiy@microsoft.com).

Z. Ji is with QUALCOMM, San Diego, CA 92121 USA (e-mail: zhuji@glue.umd.edu).

K. J. R. Liu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@isr.umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2007.897270

[1] A Nash equilibrium is a strategy profile for a game with the property that no player can benefit by changing his or her strategy while the other players keep their strategies unchanged [19].

can be regarded as the worst-case scenario from the defenders' point of view. That is, if a strategy can work well under this scenario, they can work well under any scenario.

In general, the environment is noisy and full of uncertainty, which may consequently result in that some decisions cannot be perfectly executed. For example, even a node wants to forward a packet for another node, this packet may still be dropped due to link breakage. Further, perfect monitoring, that is, any action outcome can be correctly observed, is either impossible or too expensive to afford in ad-hoc networks due to the distributed nature and the limited resources. In this paper, the effects of noise and imperfect monitoring on the strategy design will be investigated, and the optimal defending strategies under both noise and imperfect monitoring will be devised by incorporating statistical attacker detection mechanisms. The analysis shows that the devised strategies are optimal in the sense that no other strategies can further increase good nodes' payoff under attacks. Meanwhile, the optimal attacking strategies have also been devised.

The rest of this paper is organized as follows. Section II describes the secure routing and packet forwarding game model with incomplete type information. Section III presents the devised defending strategies by incorporating statistical attacker detection mechanisms. The possible attacking strategies have also been studied in this section. The optimality analysis of the devised strategies is presented in Section IV. The justification of the underlying assumptions and the performance evaluation of the devised strategies are demonstrated in Section V. Finally, Section VI concludes this paper.

II. System Description and Game Model

A. *System Description*

In this paper, we consider cooperative ad-hoc networks deployed in adversarial environments. According to their objectives, nodes in such networks can be classified into two types: good and malicious. The objective of good nodes is to optimize the overall system performance, while the objective of malicious nodes is to maximize the damage that they can cause to the system. In such networks, each node may have some data scheduled to be delivered to certain destinations, and the data rate from each node is determined by the common system goal, which is usually application specific.

In general, due to the multihop nature, when a node wants to send a packet to a certain destination, a sequence of nodes needs to be requested to help forward this packet. We refer to the sequence of (ordered) nodes as a route, the intermediate nodes on a route as relay nodes, and the procedure to discover a route as route discovery. In general, the route discovery has three stages. In the first stage, the requester notifies the other nodes in the network that it wants to find a route to a certain destination. In the second stage, other nodes in the network will make their decisions on whether to agree to be on the discovered route or not. In the third stage, the requester will determine which route should be used.

Based on whether having gained access to the network, the attackers can be classified into two types: insider attackers and outside attackers, where the former are legitimate users who have privilege to utilize the network resources, while the latter are not legitimate and have no privilege to utilize the network resources. To defend against outside attackers, the existing schemes based on access control and secret communication channels can work well [1], [4]–[6], [10]–[14]. Defending against insider attackers, however, is more challenging due to the fully cooperative nature of such networks. In this paper, we focus on insider attackers.

In general, a variety of attacks can be launched, ranging from passive eavesdropping to active interfering. Since we focus on packet forwarding, we will mainly consider the following two general attack models: dropping packets and injecting traffic. By dropping other nodes' packets, all of the resources spent to transmit these packets are wasted, and the network's performance is degraded. Attackers can also inject an overwhelming amount of packets into the network: once the others have forwarded these packets but cannot get payback, those resources spent to forward these packets are wasted. Meanwhile, the attackers are allowed to collude to increase their attacking capability.

In cooperative ad-hoc networks, without knowing any information about the node's legitimate data generation rate, the detection of injecting traffic attacks will become extremely hard (or impossible). Fortunately, since cooperative ad-hoc networks are designed to fulfill certain common goals, it holds generally that a node's legitimate data generation rate can be known or estimated by some other nodes in the network. For example, in ad-hoc sensor networks designed to conduct environment surveillance, each node needs to send collected information to the centralized data collector, and the amount of data that each node can send is usually predetermined by the system goal, and can be known or estimated by some other legitimate nodes. In this paper, we assume that for each node s in the network, the number of packets that it will generate by time t is $T_s(t)$, which is usually different for different node.[2] In general, the exact value of $T_s(t)$ may not be known by other nodes in the network. In this paper, we assume that the upperbound of $T_s(t)$, denoted by $f_s(t)$, can be known or estimated by some nodes in the network.

In wireless ad-hoc networks, some decisions may not be perfectly executed. For example, when a node has decided to help another node to forward a packet, the packet may still be dropped due to link breakage or transmission errors.[3] In this paper, we consider the following decision execution error, that is, the decision is to forward a packet but the outcome is the packet being dropped. Meanwhile, in such wireless networks, each node only has a local, private, and imperfect observation about the other nodes' behavior. Specifically, even a packet has been successfully forwarded; this can still be observed as packet dropping by some nodes (e.g., this may occur frequently when watchdog [3] is used). Similarly, a dropping packet event can also be observed as packet forwarding.

[2] In general, the number of packets that each node s will generate by time t can be modeled as a random variable, and $T_s(t)$ can be regarded as a specific realization.

[3] We refer to those factors causing imperfect decision execution as noise, which may include environmental unpredictability and system uncertainty, channel errors, mobility, congestion, etc.

B. Game Model

To formally analyze the security issue in cooperative ad-hoc networks, we model the dynamic interactions between good nodes and attackers as a **secure routing and packet forwarding game** with incomplete type information and imperfect observation.

- **Players:** The set of players is denoted by N, which is the set of legitimate nodes in the network.
- **Types:** Each player $i \in N$ has a type $\theta_i \in \Theta$ where $\Theta = \{good, malicious\}$. Initially, each attacker knows any other player's type, while each good player assumes all nodes are good. That is, good nodes have incomplete information of the others' type. Let N_g denote the set of good players and $N_m = N - N_g$ the set of attackers.
- **Strategy space:**[4]
  1) **Route participation stage**: For relay node, after receiving a message requesting it to be on a certain route, it can either accept this request, denoted by A, or not accept this request, denoted by NA.
  2) **Route selection stage**: For each source node who has a packet to send, after discovering a valid route,[5] its decision can be either request/use this route to send the packet, denoted by R, or not request/use this route to send the packet, denoted by NR.
  3) **Packet forwarding stage**: For each relay node, once it has received a packet requesting it to forward, its decision can be either to forward this packet, denoted by F, or drop this packet, denoted by D.
- **Cost:** For any player $i \in N$, transmitting a packet, either for itself or for the others, will incur cost c_i.
- **Gain:** For each good player $i \in N_g$, if a packet originated from it can be successfully delivered to its destination, it can get gain g_i.
- **Imperfect execution:** Due to noise, with probability p_e, each decision F can be mistakenly executed as D.
- **Imperfect observation:** With probability p_m, each forwarding outcome can be observed as dropping by the source (i.e., miss probability), and with probability p_f, each dropping outcome can be observed as forwarding by the source (i.e., false positive probability). Meanwhile, when a node has injected a packet, with probability p_s, it can avoid being detected by those who know its legitimate traffic injection rate.
- **Utility:** For each player $i \in N$, we can model the players' payoff functions as follows (cf. Table I for notations).
  1) **Good players:** Since all good players belong to the same authority and pursue common goals, they will share the same utility function as follows:

$$U_g(t_f) = \frac{\sum_{i \in N_g} \left(S_i(t_f) g_i - F_i(t_f) c_i\right)}{\sum_{i \in N_g} T_i(t_f)} \tag{1}$$

TABLE I
SUMMARY OF NOTATIONS

p_m	the probability that a forwarding outcome is observed as dropping (i.e., miss detect probability)
p_f	the probability that a dropping outcome is observed as forwarding (i.e., false positive probability)
p_s	the probability that a node can avoid being detected when injected a packet
p_e	the probability that a forwarding decision can be mistakenly executed as dropping
g_i	the gain to node i if a packet originated from it can be successfully delivered to its destination
c_i	the cost incurred to node i to transmit a packet
t_f	the lifetime of the network
L_{max}	a pre-determined system parameter to specify the maximum number of hops per route.
$\bar{L}_{min}$	the average number of hops per selected route
$T_s(t)$	the number of packets that it will generate by time t
$f_s(t)$	the upper-bound of $T_s(t)$ can be known or estimated by some nodes in the network
$S_i(t)$	the number of i's packets that have been scheduled to send and have successfully arrived at their destinations by time t
$F_i(j,t)$	the number of packets that i has forwarded for player $j \in N$ by time t
$W_i(j,t)$	the total times of wasted packet transmissions that i has caused to j by time t due to i dropping those packets that have been transmitted by j
$R_i(j,t)$	denote the number of times that node j has agreed to forward for node i by time t.
$\tilde{F}_j(i,t)$	the number of times that i has observed that j has forwarded a packet for it by time t
$\tilde{T}_j(t)$	the number of packets that have been injected by j and have been observed by those nodes who know j's legitimate traffic injection rate

where

$$F_i(t) = \sum_{j \in N} F_i(j,t). \tag{2}$$

  2) **Malicious players:** Since malicious players are allowed to collude, we assume they will also share the same utility function, defined as follows:

$$U_m(t_f) = \frac{\sum_{i \in N_m} \left(\sum_{j \in N_g} (W_i(j,t_f) + F_j(i,t_f)) c_j - \alpha F_i(t_f) c_i \right)}{t_f}. \tag{3}$$

Here, parameter α is introduced to determine the relative importance of attackers' cost comparing to good players' cost. That is, from the attackers' point of view, it is worth spending cost c to cause the damage worth c' to good players as long as $\alpha < c'/c$.

The objective of good players is to maximize U_g, while the objective of attackers is to maximize U_m. If the game will be played for an infinite duration, then their utility functions will become $U_g = \lim_{t \to \infty} U_g(t)$ and $U_m = \lim_{t \to \infty} U_m(t)$, respectively.

Remark 1: On the right-hand side of (1), the numerator denotes the net profit (i.e., total gain minus total cost) that the good nodes obtained, and the denominator denotes the total number of packets that good nodes need to send. The utility function (1) represents the average net profit that good nodes can obtain per packet that needs to be delivered. Since good nodes do not have any prior knowledge of the other nodes' types, each good node

[4] Each node can act both as a source and as a relay, and has different strategy spaces when acting as different roles.

[5] A valid route means that all nodes on this route have agreed to be on this route and each node on this route lies inside the transmission range of its previous player on this route.

may not know its exact payoff by time t, which introduces extra difficulty to optimal strategy design.

Remark 2: In (3), $W_i(j,t)c_j$ represents the total damage (or wasted energy) that i has caused to j by time t due to i launching dropping packets attacks, $F_j(i,t)c_j$ represents the total damage that i has caused to j by time t due to i launching injecting traffic attacks, and $F_i(t)c_i$ represents the total cost incurred to i by launching both injecting traffic and dropping packets attacks by time t. In summary, the numerator of the right-hand side of (3) represents the net damage that the attackers caused to the good nodes. Since this value may increase monotonically, it is normalized by dividing the network lifetime t_f. Now this utility function represents the average net damage that the attackers cause to the good nodes per time unit. From (3), we can see that in this game setting, the attackers' goal is to waste the good nodes' energy as much as possible. Alteratively, attackers can also have other types of goals, such as minimizing the good nodes' payoff. In Section IV, we will show that the performance of the proposed defending strategy is not sensitive to the attackers' specific goal and, in most situations, maximizing (3) has the same effect as minimizing the good nodes' payoff under the proposed defending strategies.

Remark 3: The above game can be divided into many subgames as explained below. Once a player wants to send a packet to a certain destination, a subgame will be initiated which consists of three stages: in the first stage, the source will request some players to be on a certain route to the destination; in the second stage, the source will decide whether it should use this route to send the packet; in the third stage, each relay will decide whether it should help the source to forward this packet once it has received it.

To simplify our illustration, we assume that $g_i = g$ for all $i \in N_g$ and $c_i = c$ for all $i \in N$. Like many other routing protocols for ad-hoc networks in the above game, the maximum number of hops per route will be upperbounded by $L_{\max}$, which is a predetermined system parameter. Without loss of generality, we assume that $(1-p_e)^{L_{\max}} g > L_{\max} c$; otherwise, the expected gain may be less than the expected cost. Since in ad-hoc networks, energy is usually the most precious resource, we can directly relate the cost to energy. The physical meaning of gain g may vary according to specific applications. However, as to be shown in Section IV, as long as g is reasonably large, it will not affect the strategy design.

According to the above game model, in each single routing and packet forwarding subgame, for the initiator of this subgame, its strategy space is {R, NR}, while for each relay node, its strategy space is {(A, F), (A, D), (NA, F), (NA, D)}. Here, (A, F) means that a relay node agrees to be on a certain route in the route participation stage and will forward the packet from the source in the packet forwarding stage, (A, D) means that a relay node agrees to be on a certain route in the route participation stage but will drop the packet from the source in the packet forwarding stage, (NA, F) means that a relay node does not agree to be on a certain route but will forward the packet from the source in the packet forwarding stage, and (NA, D) means that a relay node does not agree to be on a certain route and will drop the packet from the source in the packet forwarding stage.

In the above game, we have assumed that some necessary monitoring mechanisms will be launched to detect possible packet dropping, such as those described in [3], [8], [20], and [21]. We have also assumed that when a node transmits a packet, its neighbors can know who the source of this packet is and who is currently transmitting this packet. This can be achieved by the monitoring mechanism described in [9]. However, we do not assume any perfect monitoring, and each node makes its decision only based on local private and imperfect observation. In general, p_f, p_m, and p_s are determined by the underlying monitoring mechanism.

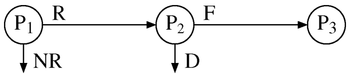


Fig. 1. Single routing and packet forwarding subgame.

In this paper, we proposed a set of strategies to secure cooperative ad-hoc network against insider attackers under noise and imperfect observation. However, it is also worth pointing out that in order for the proposed strategies to work well, the existing security schemes, such as those described in [1], [4]–[6], [10]–[14], should also be incorporated to achieve necessary access control, authentication, data integrity, and so on. In general, besides dropping packet, injecting traffic, and collusion, there also exist a variety of other types of attacks, such as jamming, slander, etc. A main contribution of this paper is to provide some insight on securing ad-hoc networks under noise and imperfect observation. To our best knowledge, this paper is the first one to provide a game theoretical analysis of securing routing and packet forwarding in cooperative ad-hoc networks under noise and imperfect monitoring.

III. Defense Strategies With Statistical Attacker Detection

We first briefly study a simple subgame with complete type information and perfect observation: P_1 requests P_2 to forward a packet to P_3 through the route "$P_1 \to P_2 \to P_3$", and P_2 has agreed to be on this route, as illustrated in Fig. 1. Since the type information is complete, all players know each other's type. This is a two-stage extensive game with P_1 moving first. If P_1's action is NR, then the game will be terminated immediately; otherwise, P_2 will take its action accordingly. The payoff profiles for this game under different scenarios are shown in Fig. 2, where the first value in each payoff profile corresponds to P_1's payoff and the second corresponds to P_2's payoff. Here, payoff is defined as the gain minus the cost in this subgame. Based on the types of P_1 and P_2, there are four different scenarios.

- Scenario 1: P_1 is good and P_2 is bad. Then the only Nash equilibrium is (NR, D) with payoff profile (0, 0), since no one can further increase their payoff by deviating.
- Scenario 2: P_1 is bad and P_2 is good. Then, the only Nash equilibrium is (NR, D) with payoff profile (0, 0).
- Scenario 3: Both players are good. In this scenario, if $g > 2c$, the only Nash equilibrium is (R, F) with payoff profile $(g-2c, g-2c)$; if $g < 2c$, the only Nash equilibrium is (NR, F) with payoff profile (0, 0); while if $g = 2c$, there are two Nash equilibria (NR, F) and (R, F), both have the same payoff profile (0, 0).

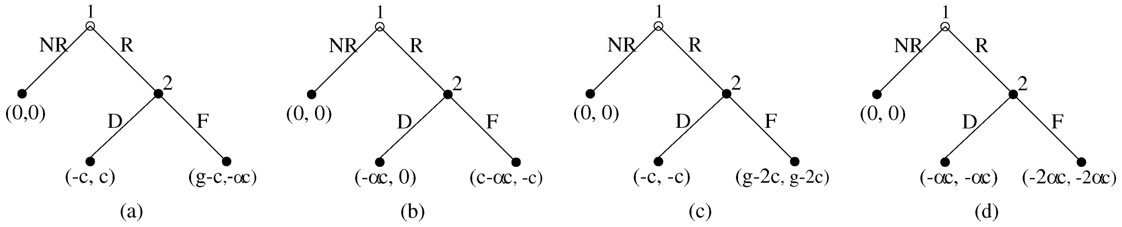


Fig. 2. Payoff profiles under different scenarios. (a) P_1 is good, P_2 is bad. (b) P_1 is bad, P_2 is good. (c) Both players are good. (d) Both players are bad.

- Scenario 4: Both players are bad. Then the only Nash equilibrium is (NR, D) with payoff profile (0, 0).

Based on the above analysis, we can conclude that in a two-hop subgame with complete type information.

1) A good node should neither forward any packet for attackers nor request any attackers to forward packets. Meanwhile, good nodes should always forward packets for other good nodes provided $g > 2c$.
2) A malicious node should not forward any packet and should not request other nodes to forward packets.

This can be easily generalized to the multihop scenario, that is, no good nodes should work with malicious nodes.

However, defending against insider attacks in realistic scenarios is much more challenging due to the following reasons. First, good nodes cannot know who are attackers *a priori*. Second, owing to noise, the decision execution may not be perfect. Third, monitoring errors will be very common because of the fully distributed nature and limited available resources. Consequently, the attackers can easily take advantage of such information asymmetry and imperfectness to cause more damage and avoid being detected.

To handle incomplete type information, certain attacker detection mechanisms should be applied. In general, one can base on what being observed to detect malicious behavior. For example, if a node has agreed to forward a packet but later drops it, other nodes (either its neighbor or the source of the packet) who have observed this inconsistency (i.e., agreeing to forward but dropping) can mark this node as malicious. If there is no decision execution error and the observation is perfect, such a method can detect all intentional packet dropping.

However, noise always exists and the monitoring is impossible to be perfect. Under such realistic circumstances, detecting malicious behavior will become extremely hard due to that an observed misbehavior may either be caused by intention, or by unintentional execution error, or simply due to observation error. Now a node should not be marked as malicious just simply because it has been observed dropping some packets. Accordingly, the attackers can take advantage of noise and observation errors to cause more damage without being detected.

A. Statistical Dropping Packet Attack Detection

To combat insider attacks under noise and imperfect observation, we first study what should be normal observation when no attackers are present. In this case, when a node has made a decision to forward a packet (i.e., decision F), the probability p_F that the outcome observed by the source is also forwarding can be calculated as follows:

$$p_F = (1 - p_e)(1 - p_f) + p_e p_m. \tag{4}$$

Let $R_i(j,t)$ denote the number of times that node j has agreed to forward for node i by time t, and $\tilde{F}_j(i,t)$ denote the number of times that i has observed that j has forwarded a packet for it by time t. Based on the central limit theorem (CLT) [22], for any $x \in \mathcal{R}^+$, we can have

$$\lim_{R_i(j,t)\to\infty} \text{Prob}\left(\frac{\tilde{F}_j(i,t) - R_i(j,t)\cdot p_F}{\sqrt{R_i(j,t)\cdot(1-p_F)\cdot p_F}} \geq -x\right) = \Phi(x) \tag{5}$$

where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt. \tag{6}$$

In other words, when $R_i(j,t)$ is reasonably large, $\tilde{F}_j(i,t) - R_i(j,t)p_F$ can be approximately modeled as a Gaussian random variable with mean 0 and variance $R_i(j,t)p_F(1-p_F)$.

Let $isDPA_i(j)$ denote i's belief about whether j has launched dropping packets attack, where $isDPA_i(j) = 1$ indicates that i believes j has launched dropping packets attack, while $isDPA_i(j) = 0$ indicates that i believes j has not launched dropping packets attack. Let B_{th} be a reasonably large constant (e.g., 200). Then, the following hypothesis testing rule can be used by i to judge whether j has maliciously dropped its packets: See (7) at the bottom of the page. If (7) is used to detect the dropping packets attack, the false alarm ratio would be no more than $1 - \Phi(x)$. It is worth mentioning that even for a small positive x, the value of $\Phi(x)$ can still approach 1 (e.g, $\Phi(5) > 0.999$).

$$isDPA_i(j) = \begin{cases} 1, & \text{if } \tilde{F}_j(i,t) < R_i(j,t)p_F - x\sqrt{R_i(j,t)p_F(1-p_F)} \text{ and } R_i(j,t) > B_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

B. Statistical Injecting Traffic Attack Detection

In Section III-A, we focus on dropping packets attacks. Attackers can also try to inject an overwhelming amount of traffic to waste the good nodes' resources. Let $isITA_i(j)$ denote i's belief about whether j has launched injecting traffic attack, where $isITA_i(j) = 1$ indicates that i believes j has launched injecting traffic attack, while $isITA_i(j) = 0$ indicates that i believes j has not launched an injecting traffic attack. Let $\tilde{T}_j(t)$ denote the number of packets that have been injected by j and have been observed by those nodes who know j's legitimate traffic injection rate. Then, a simple detection rule can be as follows:

$$isITA_i(j) = \begin{cases} 1, & \text{if } \tilde{T}_j(t) > f_j(t) \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Under this detection rule, the maximum number of packets that attacker j can inject without being detected will be no more than $f_j(t)/(1-p_s)$. This detection rule is very conservative since only those observed packet injection events are used. If p_s can also be known by good nodes, we can modify (8) to further limit the number of packets that j can inject without being marked as malicious, such as changing the threshold from $f_j(t)$ to $f_j(t)/(1-p_s)$. Since p_s is usually not known and may change across different nodes, in this paper, when performing injecting traffic attack detection, we will not incorporate p_s into the detection rule.

The detection rule (8) can work well only when no retransmission is allowed. Next, we show how to detect injecting traffic attack when retransmission is allowed upon unsuccessful delivery. We first make a simple assumption that all selected routes have the same number of hops, denoted by L, and let $q = (1-p_e)^L$. Then, for each packet, the total number of tries needed to successfully deliver this packet to its destination can be modeled as a geometric random variable with mean $1/q$ and variance $(1-q)/q^2$. For any node $j \in N$, if $p_s = 0$ and j have never intentionally retransmitted a packet that has been successfully delivered to its destination, according to the CLT, for any $x \in \mathcal{R}^+$, we should have

$$\lim_{\hat{T}_j(t)\to\infty} \text{Prob}\left(\frac{\tilde{T}_j(t) - \frac{T_j(t)}{q}}{\sqrt{\frac{T_j(t)(1-q)}{q^2}}} \leq x\right) = \Phi(x). \tag{9}$$

In other words, when $\tilde{T}_j(t)$ is reasonably large, $\tilde{T}_j(t) - T_j(t)/q$ can also be approximately modeled as a Gaussian random variable with mean 0 and variance $T_j(t)(1-q)/q^2$. Then, a modified detection rule can be as follows:

$$isITA_i(j) = \begin{cases} 1, & \text{if } \tilde{T}_j(t) > \frac{f_j(t)}{q} + \frac{x\sqrt{f_j(t)(1-q)}}{q} \text{ and } \tilde{T}_j(t) > B_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Similarly, when the above detection rule is used, the false alarm ratio would be no more than $1 - \Phi(x)$. In this case, the number of packets that attacker j can inject without being marked as malicious is upperbounded by $(f_j(t) + x\sqrt{f_j(t)(1-q)})/q(1-p_s)$. Comparing to the case that no retransmission is allowed, when retransmission is allowed, attackers can inject more packets without being detected, though good nodes can also enjoy higher throughput.

In general, the number of hops per selected route varies according to the locations of the source and destination and the network topology. Let $\bar{L}_{\min}$ denote the average number of hops per selected route. When calculating q used in (10), an alternative way is to let $q = (1-p_e)^{\bar{L}_{\min}}$. However, this may lead to higher false alarm probability since some nodes may experience longer routes due to their locations. In this paper, we adopt a more conservative way by letting $q = (1-p_e)^{L_{\max}}$. As a consequence, even when $p_s = 0$, the resulting false positive probability will be far less than $1 - \Phi(x)$, with the penalty that the attackers can also inject more packets without being detected. For example, for $L_{\max} = 10$, $\bar{L}_{\min} = 4$, $p_e = 0.02$, the extra increase would be about 12.9% (i.e., $(1-p_e)^{\bar{L}_{\min} - L_{\max}} - 1$). Accordingly, the good nodes' payoff will also be decreased.

C. Secure Routing and Packet Forwarding Strategy

Based on the above analysis, we can arrive at the following strategy to secure routing and packet forwarding in cooperative ad-hoc networks under noise and imperfect monitoring:

1) Secure Routing and Packet Forwarding Strategy: In the secure routing and packet forwarding game under noise and imperfect monitoring, initially each good node will assume all other nodes are good. For each single routing and packet forwarding subgame, assuming that P_0 is good and is the source who wants to send a packet to P_n at time t, and a route "$P_0 \to P_1 \to \cdots \to P_n$" has been discovered by P_0. After P_0 has sent requests to all the relays on this route asking them to participate, for each good node on this route the following strategies should be taken in different stages.

1) In the route participation stage: A good relay P_i takes action A if and only if no nodes on this route have been marked as bad and $n \leq L_{\max}$;[6] otherwise, it takes NA.
2) In the route selection stage: P_0 will take action R if and only if all of the following conditions can be satisfied: 1) the packet is valid (i.e., it is scheduled to be sent by P_0); 2) $n \leq L_{\max}$; 3) no nodes on this route have been marked as malicious by P_0; 4) all relays have agreed to forward packets in the route participation stage; and 5) this route has the minimum number of hops among all good routes to P_n known by P_0; otherwise, P_0 should take action NR.
3) In the packet forwarding stage: For each relay P_i, it will take action F if and only if it has agreed to be on this route in the route participation stage; otherwise, it should take action D.

Let x be a positive constant. For any node j, it will be marked as malicious by node i if it has been detected by any following rules (7) and (8) if retransmission is not allowed, and (10) if retransmission is allowed, where in (10) $q = (1-p_e)^{L_{\max}}$. Meanwhile, node j will also be marked as malicious if it has requested to send a packet through a route with the number of hops greater than $L_{\max}$.

In the above defense strategy, each good node needs to know or estimate the following parameters p_e, p_f, p_m, and $L_{\max}$. Meanwhile, it also needs to set the two constants that are used

[6] $n \leq L_{\max}$ leads to that the expected cost is less than the expected gain.

in (7) and (10): B_{th} and x. $L_{\max}$ is a system-level parameter and is known by all nodes in the network. The packet dropping probability p_e can be either trained offline, or estimated online by each node through evaluating its own packet dropping ratio. In general, a different node may experience different p_e at a different time or location. Under such circumstances, to reduce the false positive when performing attacker detection using (7) and (10), a node may set p_e to be a little bit larger than the one experienced by itself. The two observation error-related parameters p_f and p_m can be provided by the underlying monitoring mechanism. Similarly, a different node may also experience different p_f and p_m at different situations. Therefore, when a node uses (7) to perform attacker detection, to limit the false positive, it may use the upperbounds of p_f and p_m provided by the underlying monitoring mechanisms. This will be further studied later.

D. Attacking Strategy

Since this paper focuses on insider attackers, it is reasonable to believe that attackers can know the defending strategies employed by the system. This can be regarded as the worst-case scenario from the defenders' point of view. This subsection studies what strategies the attackers should use to maximize their payoff when the proposed secure routing and packet forwarding strategy is used by the good nodes.

We first study dropping packets attack. According to the proposed secure routing and packet forwarding strategy, once a node i has been marked as malicious by another node j, i will not be able to cause damage to j again. Therefore, an attacker should avoid being detected in order to continuously cause damage to the good nodes. A simple strategy is to always apply (A, D). However, when applying this strategy, the maximum number of good nodes' packets that an attacker can drop without being detected will be no more than $|N_g| \cdot B_{th}$, while the penalty is that it will be detected as malicious and cannot cause damage to the good nodes anymore.

Intuitively, attackers can selectively drop packets to avoid being detected and still cause continuous damage to good nodes. According to the proposed secure routing and packet forwarding strategy, the number of a good node i's packets that an attacker j can drop without being detected is upperbounded by $np_e + x\sqrt{n}/(1+p_m-p_f)$,[7] where n is the number of packets that j has agreed to forward for i. In other words, j has to forward at least $n(1-p_e) - x\sqrt{n}/(1+p_m-p_f)$ packets for i in order to avoid being marked as malicious. However, recall that even if there are no attackers, in average $n(1-p_e)$, packets will be dropped due to noise. That is, the extra number of i's packets that j can selectively drop without being detected is upperbounded by $x\sqrt{n}/(1+p_m-p_f)$, while the cost needed to forward packets for j is at least $n(1-p_e)c - x\sqrt{n}/(1+p_m-p_f)c$. Since we have $\lim_{n\to\infty}(x\sqrt{n}/(1+p_m-p_f))/n(1-p_e) = 0$, selectively dropping i's packets can bring almost no gain to j if the game will be played for a long enough time.

According to the secure routing and packet forwarding strategy, a good node will not start performing dropping packet attack detection before having enough interactions with another node (e.g., B_{th}). Therefore, the following dropping packet attack strategy can be used by an attacker j when acting as relay nodes: for each good node i, it can drop the first $B_{th}-1$ i's packets by playing (A, D), then start playing (NA, D) forever. With this strategy, the damage that j can cause to i is upperbounded by $B_{th}c$ without introducing any cost to j. It is easy to see that the relative damage $B_{th}c/t_f$ decreases monotonically with the increase of the network lifetime t_f.

Until now, we have assumed that all nodes will experience the same p_e, p_f, and p_m. However, such an assumption may not hold in general. For example, attackers may be able to decrease p_f and/or increase p_m experienced by it. Let p'_f and p'_m denote the actual false positive probability and miss probability experienced by an attacker j. When j tries to drop i's packets, in order to avoid being detected, the actual packet drop ratio p'_e that j will apply to drop i's packet should satisfy (11)

$$(1-p'_e)(1-p'_f) + p'_e p'_m > p_F - \frac{x\sqrt{np_F(1-p_F)}}{n} \tag{11}$$

where n is the number of packets that j has agreed to forward for i. It is easy to check that to satisfy (11) for all possible n, the maximum packet dropping ratio p'_e that j can apply is upperbounded by

$$p'_e \leq \frac{p_e(1-p_f-p_m) + (p_f - p'_f)}{1-p'_f-p'_m}. \tag{12}$$

From (12), we can see that increasing the miss probability p'_m and/or decreasing the false positive probability experienced by j can also increase p'_e and, consequently, increase the damage to i. Let L_{avg} denote the average number of wasted packet transmissions caused by j when it drops i's packets, according to the payoff definition (3), as long as $(p'_e - p_e)/(1-p_e) \leq \alpha L_{\text{avg}}$, launching dropping packet attack with p'_e cannot bring gain to j. However, if $(p'_e - p_e)/(1-p_e) > \alpha L_{\text{avg}}$, j should launch dropping packet attacks by selectively dropping the good nodes' packets with the dropping probability calculated based on (12).

Now we study the injecting traffic attack. According to the secure routing and packet forwarding strategy, to avoid being marked as a launching injecting traffic attack, an attacker j should be sure that $\tilde{T}_j(t) \leq f_j(t)$. However, j may not know the exact value of $\tilde{T}_j(t)$, and needs to estimate $\tilde{T}_j(t)$ by itself. Recall that for each packet injected by j, with probability p_s, it can avoid being detected. It is readily to show that $\tilde{T}_j(t) - F_j(j,t)(1-p_s)$ can be approximately modeled as a Gaussian random variable with mean 0 and variance $F_j(j,t)p_s(1-p_s)$, where $F_j(j,t)$ is the total number of packets injected by j until time t.

Based on the above analysis, when no retransmission is allowed, a good injecting traffic strategy is as follows: j should try to limit the number of injected packets $F_j(j,t)$ to satisfy the following condition:

$$F_j(j,t)(1-p_s) + y\sqrt{F_j(j,t)p_s(1-p_s)} < f_j(t) \tag{13}$$

[7] It is ready to check that

$$\left(np_e + \frac{x\sqrt{n}}{1+p_m-p_f}\right)p_m + (n(1-p_e) - \frac{x\sqrt{n}}{1+p_m-p_f})p_f$$
$$= np_F - x\sqrt{n} < np_F - x\sqrt{np_F(1-p_F)}.$$

where y is a large positive constant. By using this strategy, the probability that j will be detected is upperbounded by $1-\Phi(y)$. When retransmission is allowed, according to the secure routing and packet forwarding strategy, the condition should be changed as follows:

$$F_j(j,t)(1-p_s)+y\sqrt{F_j(j,t)p_s(1-p_s)} < \frac{f_j(t)+x\sqrt{f_j(t)(1-q)}}{q} \qquad (14)$$

where y is a large positive constant and x and q are defined in the secure routing and packet forwarding strategy.

In summary, we can arrive at the following attacking strategy, referred to as the optimal attacking strategy.

1) Dropping packet attack: For any attacker j, if the maximum possible p_e' calculated using (12) is larger than p_e and $(p_e'-p_e)/(1-p_e) \leq \alpha L_{\text{avg}}$, it should try to selectively drop the good nodes' packets with probability p_e'; otherwise, it should apply the following strategy: for any good node i, j should try to drop the first $B_{th}-1$ i's packets by playing (A, D), then start playing (NA, D) forever when acting as relay node for i.
2) Injecting traffic attack: For any attacker j, if no retransmission is allowed, it should try to inject traffic by following (13); otherwise, it should try to inject traffic by following (14). Meanwhile, when j has decided to inject a packet, it should pick a route with the following properties: 1) the number of hops is no more than $L_{\max}$; 2) all relays are good nodes; and 3) among all of the routes known by j which satisfy (a) and (b), this route has the maximum number of hops.

IV. Optimality Analysis

In this section, we analyze the optimality of the proposed strategy profile, where all good nodes follow the strategy described in Section III-C and all attackers follow the strategy described in Section III-D. We will focus on the worst case scenario from the good nodes' point of view: when a malicious node wants to send a packet to another node, it can always find a route with $L_{\max}$ hops and all relay nodes being good. This is also the best case scenario from the attackers' point of view. We focus on the scenario that all nodes experience the same p_e, p_f, and p_m. The scenario that the different node will experience different p_e, p_f, and p_m will be discussed at the end of this section.

Theorem 1: In the secure routing and packet forwarding game in noiseless environment with perfect observation (i.e., $p_e = p_f = p_m = p_s = 0$), the proposed strategy profile with $B_{th} = 1$ form a Nash equilibrium.

Proof: See the Appendix. ■

Now we analyze the nodes' possible payoff under the proposed strategy profile. Let $f_i^{\text{avg}} = f_i(t_f)/t_f$ when t_f is finite, and $f_i^{\text{avg}} = \lim_{t\to\infty} f_i(t)/t$ when t_f is infinite. According to the secure routing and packet forwarding strategy, a good node will not work with any node that has been marked as malicious by itself. First, as we have shown in Section III-D, playing (A, D) cannot increase the attackers' payoff provided t_f is infinite. Second, it is easy to see that playing (NA, F) and (A, F) cannot increase the attackers' payoff either, since when an attacker plays (NA, F), no good nodes will request it to forward packets, while when an attacker plays (A, F), it can only make contribution to the good nodes. Third, when an attacker tries to inject packets, similar to the analysis in the proof of Theorem 1, it should always use the route with all relay nodes being good and having agreed to be on the route. Meanwhile, from an attacker's point of view, injecting more packets than specified will make it to be marked as malicious and cannot cause any more damage to the good nodes, and consequently decrease its payoff. Therefore, when no retransmission is allowed, based on (3), the attackers' payoff will be upper-bounded by

$$\begin{aligned} U_m &\leq \lim_{t_f\to\infty}\frac{1}{t_f}\sum_{i\in N_m}\left(\frac{f_i(t_f)}{1-p_s}(L_{\max}-1-\alpha)c+|N_g|B_{th}L_{\text{avg}}c\right) \\ &= \sum_{i\in N_m}\frac{f_i^{\text{avg}}}{1-p_s}(L_{\max}-1-\alpha)c. \end{aligned} \qquad (15)$$

Here, $f_i(t_f)/(1-p_s)$ is the number of packets that attacker i can inject into the network by time t_f without being marked as malicious, $(L_{\max}-1)c$ is the maximum possible damage that can be caused to good nodes by launching injecting traffic attacks, αc is the cost incurred to attackers by forwarding a packet, and $|N_g|B_{th}L_{\text{avg}}c$ is the damage that j can cause by launching dropping packet attack.

When retransmission is allowed upon unsuccessful delivery, from the attackers' point of view, the only difference is that they can inject more packets without being detected. Now the attackers' payoff will be upperbounded by (16), shown at the bottom of the page.

Now we analyze the good nodes' payoff. Recall that $\bar{L}_{\min}$ denotes the average number of hops among those routes selected by good nodes. We first consider the situation that the environment is noisy and no retransmission is allowed. In this case, some good nodes' packets will be dropped due to noise, and $\lim_{t\to\infty} S_i(t)/T_i(t) = (1-p_e)^{\bar{L}_{\min}}$. According to (1), for each $i \in N_g$, $F_i(t)$ comes from two parts: forwarding packets for the good nodes and forwarding packets for the attackers. The total number of packets that the good nodes have forwarded for themselves is $\sum_{i\in N_g} T_i(t)\bar{L}_{\min}$ by time t, and the total

$$\begin{aligned} U_m &\leq \lim_{t_f\to\infty}\frac{1}{t_f}\sum_{i\in N_m}\left(\left(\frac{f_i(t_f)}{(1-p_s)q}+\frac{x\sqrt{f_i(t_f)(1-q)}}{q}\right)(L_{\max}-1-\alpha)c+|N_g|B_{th}L_{\text{avg}}c\right) \\ &= \sum_{i\in N_m}\frac{f_i^{\text{avg}}}{(1-p_s)q}(L_{\max}-1-\alpha)c. \end{aligned} \qquad (16)$$

number of packets that the good nodes have forwarded for the attackers is no more than $\sum_{i\in N_m}(f_i(t)/(1-p_s))(L_{\max}-1)$. Meanwhile, for any given positive value x adopted in the secure routing and packet forwarding strategy, the overall false positive probability will be upperbounded by $1-\Phi(x)$, that is, at the most, $1-\Phi(x)$ percentage of good nodes will be mistakenly marked as malicious. Let $T_i^{\text{avg}} = T_i(t_f)/t_f$ when t_f is finite and $T_i^{\text{avg}} = \lim_{t\to\infty} T_i(t)/t$ when t_f is infinite. Then the good nodes' payoff will be lowerbounded by

$$\begin{aligned} U_g & \\ \geq & \lim_{t\to\infty} \frac{\sum\limits_{i\in N_g}(S_i(t)g - T_i(t)\bar{L}_{\min}c) - \sum\limits_{j\in N_m}\frac{f_j(t)}{(1-p_s)}(L_{\max}-1)c}{\sum\limits_{i\in N_g} T_i(t)} \\ = & \Phi(x)g(1-p_e)^{\bar{L}_{\min}} - \left(\bar{L}_{\min} + \frac{\sum\limits_{j\in N_m} f_j^{\text{avg}}\cdot(L_{\max}-1)}{(1-p_s)\sum\limits_{j\in N_g} T_i^{\text{avg}}}\right)c. \end{aligned} \tag{17}$$

When the environment is noiseless or when the retransmission is allowed, all good nodes' packets can be successfully delivered to their destinations with $\lim_{t\to\infty} S_i(t)/T_i(t) = 1$ for $i \in N_g$. Meanwhile, the total number of packets that the good nodes have forwarded for themselves by time t is no more than $\sum_{i\in N_g}(T_i(t)/(1-p_e)^{\bar{L}_{\min}})\bar{L}_{\min}$, and the total number of packets that the good nodes have forwarded for the attackers is no more than $\sum_{i\in N_m}(f_i(t)/q(1-p_s))(L_{\max}-1)$. Thus, in this case, the good nodes' payoff can be lowerbounded by

$$U_g \geq \Phi(x)g - \left(\frac{\bar{L}_{\min}}{(1-p_e)^{\bar{L}_{\min}}} + \frac{\sum\limits_{j\in N_m} f_j^{\text{avg}}\cdot(L_{\max}-1)}{q(1-p_s)\sum\limits_{j\in N_g} T_i^{\text{avg}}}\right)c. \tag{18}$$

On the other hand, when the proposed optimal attacking strategy is used by attackers, from the good nodes' point of view, when no retransmission is allowed, the maximum possible payoff can also be upperbounded by

$$U_g \leq g(1-p_e)^{\bar{L}_{\min}} - \left(\bar{L}_{\min} + \frac{\sum\limits_{j\in N_m} f_j^{\text{avg}}\cdot(L_{\max}-1)}{(1-p_s)\sum\limits_{j\in N_g} T_i^{\text{avg}}}\right)c. \tag{19}$$

While when retransmission is allowed, the maximum possible payoff can also be upperbounded by

$$U_g \leq g - \left(\frac{\bar{L}_{\min}}{(1-p_e)^{\bar{L}_{\min}}} + \frac{\sum\limits_{j\in N_m} f_j^{\text{avg}}\cdot(L_{\max}-1)}{q(1-p_s)\sum\limits_{j\in N_g} T_i^{\text{avg}}}\right)c. \tag{20}$$

From the above payoff analysis, we can see that the good nodes' payoff can be lowerbounded by a certain value, no matter what strategies the attackers use and what kind of goals the attackers have. In other words, the attackers' goal has little effect on good nodes' payoff when the proposed secure routing and packet forwarding strategy is used by good nodes. From the above payoff analysis, we can also see that as long as the gain g is reasonably large, it will not play an important role in the strategy design.

Theorem 2: In the infinite duration secure routing and packet forwarding game under noise and imperfect observation, the proposed secure routing and packet forwarding strategy is asymptotically optimal from the good nodes' point of view in the sense that for any $\epsilon > 0$, we can always find a $x^* > 0$ such that no other equilibrium strategies can further increase the good nodes' payoff by more than ϵ as long as the attackers also play optimally.

Proof: We first consider the situation that no retransmission is allowed. Based on the above analysis, we can see that from the attackers' point of view, to maximize their payoff, the optimal attacking strategy is to inject no more packets to the network than they are allowed and will not forward any packet for the good nodes. In this case, the good nodes' maximum possible payoff is defined in (19). According to (17), the difference between the actual payoff and maximum possible payoff is $(1-\Phi(x))(1-p_e)^{\bar{L}_{\min}}g$. Since $\Phi(x) \to 1$ as $x \to \infty$, for any $\epsilon > 0$, we can always find a constant x^* such that the actual payoff is within ϵ of the maximum possible payoff. Similarly, we can also prove this under the situation that retransmission is allowed. ■

Theorem 3: In the infinite duration secure routing and packet forwarding game, the proposed strategy profile is strongly Pareto optimal.[8]

Proof: To show the proposed strategy profile is strongly Pareto optimal, we only need to show that no other strategy profiles can further increase some players' payoff without decreasing any other player's payoff.

We first show that the good nodes' payoff cannot be further increased without decreasing the attackers' payoff. According to (17), to further increase the good nodes' payoff, one can either decrease $\bar{L}_{\min}$, or decrease f_j^{avg}. First, since the minimum-hop routes have been used, $\bar{L}_{\min}$ cannot be further decreased. Second, according to (3) and (15), decreasing f_j^{avg} always decreases the attackers' payoff.

Next, we show that the attackers' payoff cannot be further increased without decreasing the good nodes' payoff. According to (3), to increase the attackers' payoff, one can either try to increase $\sum_{i\in N_m, j\in N_g} W_i(j,t)$ and $\sum_{i\in N_m, j\in N_g} F_j(i,t)$, or try to decrease $\sum_{i\in N_m} F_i(t)$. First, $\sum_{i\in N_m, j\in N_g} F_j(i,t)$ comes completely from injecting traffic attacks, which has been maximized and cannot be further increased. Since $\sum_{i\in N_m, j\in N_g} W_i(j,t)$ comes from launching dropping packet attacks, increasing $\sum_{i\in N_m, j\in N_g} W_i(j,t)$ will also decrease the good players' payoff. Now we consider $\sum_{i\in N_m} F_i(t)$. According to the above packet forwarding strategy, attacker i will not forward packets for others, so $F_i(t)$ comes totally from transmitting packets for itself. Therefore, $F_i(t)$ cannot be further decreased without decreasing the attackers' payoff. ■

Until now, we have focused on the scenario that p_e, p_f, and p_m are kept the same for all nodes at all times. However, as we

[8] A strategy profile is said to be Pareto optimal if there is no other strategy profile which can simultaneously increase all players' payoff; a strategy profile is said to be strongly Pareto optimal if there is no other strategy profile which can increase at least one player' payoff without decreasing any other players' payoff [19].

have mentioned, this may not hold in general. Next, we study the consequence when different nodes may experience different p_e, p_f, and p_m. First, from the good nodes' point of view, such variation may increase false positive probability when performing attacker detection. For example, for a node experiencing a lower packet dropping ratio, when it uses this ratio to perform dropping packet attacker detection, with much higher probability, those nodes experiencing higher packet dropping ratio can be mistakenly marked as malicious [e.g., higher than $1-\Phi(x)$]. As mentioned in Section III-C, to avoid high false positive probability, a good node may need to set a higher p_e than the one experienced by itself when performing attacker detection. Meanwhile, a good node may also need to increase B_{th} and x to handle a possible bursty packet dropping effect, which is normal in wireless networks due to fading. Similarly, when nodes experience different p_f and p_m, a good node may need to use the upperbounds of p_f and p_m to avoid high false positive probability when performing attacker detection. As a penalty, these variations can be taken advantage of by attackers to inject more packets and drop more packets without being marked as malicious, which consequently leads to the decrease of good nodes' performance. However, our simulation studies indicate that even in such realistic scenarios, the proposed secure routing and packet forwarding strategy can still work very well.

V. Performance Evaluation

We have conducted a series of simulations to evaluate the performance of the proposed strategies in both static and mobile ad-hoc networks. In each ad-hoc network, nodes are randomly deployed inside a rectangular area of 1000 m × 1000 m. For mobile ad-hoc networks, nodes move randomly according to the random waypoint model [23], which can be characterized by the following three parameters: the average pause time, the maximum velocity $v_{\max}$, and the minimum velocity $v_{\min}$. The following physical-layer model is used: two nodes can directly communicate with each other only if they are in each other's transmission range, but it can be easily extended to a more realistic model where the error probability is a function of distance. The MAC layer protocol simulates the IEEE 802.11 distributed coordination function with a four-way handshaking mechanism [24]. Based on the above models, the static ad-hoc networks can be regarded as the noiseless case, while the mobile ad-hoc networks can be regarded as the noisy case where the decision execution error (i.e., the decision is F but the outcome is D) is only caused by link breakage. For each node, the transmission power is fixed, and the maximum transmission range is 200 m.

In the simulations, each good node will randomly pick another good node as the destination. Similarly, each attacker will also randomly pick another attacker as the destination. In both cases, packets are scheduled to be sent to this destination according to a constant rate. The total number of good nodes is set to be 100 and the total number of attackers varies from 0 to 40. For each good or malicious node, the average packet inter-arrival time is 1 s, that is, $T_i(t)=\lfloor t\rfloor$ for any time t and any node $i\in N$. Further, each good node $i\in N_g$ will set $f_i(t)=\lfloor t\rfloor+2$ for any other node $i\in N$. All data packets have the same size.

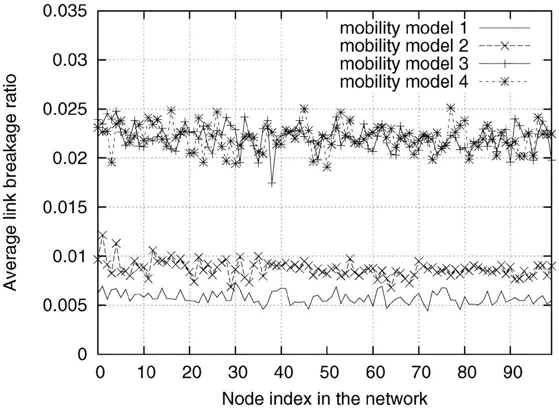


Fig. 3. Average link breakage ratio in mobile ad-hoc networks.

TABLE II
Mobility Patterns

Pattern 1:	$v_{max}=10$m/s, $v_{min}=1$m/s, pause time = 500 s
Pattern 2:	$v_{max}=15$m/s, $v_{min}=5$m/s, pause time = 300 s
Pattern 3:	$v_{max}=15$m/s, $v_{min}=5$m/s, pause time = 100 s
Pattern 4:	$v_{max}=30$m/s, $v_{min}=10$m/s, pause time = 100 s

Since the link breakage ratio p_e plays an important role in the strategy design, we first study the characteristics of link breakages in mobile ad-hoc networks under different mobility patterns. In this set of simulations, only good nodes will be considered. For each node, the average link breakage ratio experienced by it is calculated as the ratio between the total number of link breakages it experienced as the transmitter and the total number of packet transmissions it has tried as the transmitter. The total simulation time is 30 000 s. Fig. 3 shows the link breakage ratios experienced by different nodes under four different mobility patterns listed in Table II. First, from these results, we can see that the average link breakage ratio will change under different mobility patterns. Second, under the same mobility pattern, the average link breakage ratio experienced by each node is almost the same.

Fig. 4(a)–(c) shows the evolution of the average link breakage ratios over time when mobility pattern 4 is used. In this set of simulations, two nodes are randomly selected among the 100 nodes in the network. Fig. 4(a) shows the link breakage ratio averaged over every 100 s, Fig. 4(b) shows the link breakage ratio averaged over every 1000 s, and Fig. 4(c) shows the accumulated average link breakage ratio. From these results, we can see that the link breakage ratio experienced by each node may vary dramatically in a short period, but will become stable in a long period. These results suggest that when performing attacker detection, if t_f is not large enough, p_e should be set higher than the long-term average to avoid high false positive probability, while if t_f is large or goes to infinity, the average link breakage ratio can be used when performing attacker detection, with a reasonably large B_{th}.

Now we study the performance of the proposed strategies in different scenarios. We use "noiseless scenario" to denote static ad-hoc networks, and use the "noisy scenario" to denote mobile ad-hoc networks. In both cases, all good nodes follow the secure routing and packet forwarding strategy described in

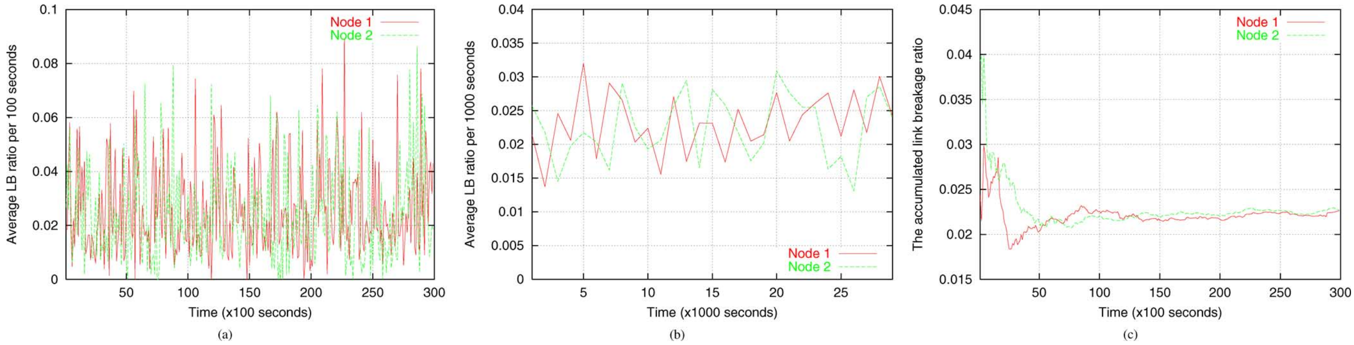


Fig. 4. Evolution of p_e in mobile ad-hoc networks.

TABLE III
NOISY SCENARIOS

Scenario 1:	Retransmission is allowed, and attackers can always find a L_{max}-hop route with all relays good.
Scenario 2:	No retransmission is allowed, and attackers can always find a L_{max}-hop route with all relays good.
Scenario 3:	Retransmission is allowed, and attackers may not find a L_{max}-hop route with all relays good.
Scenario 4:	No retransmission is allowed, and attackers may not find a L_{max}-hop route with all relays good.

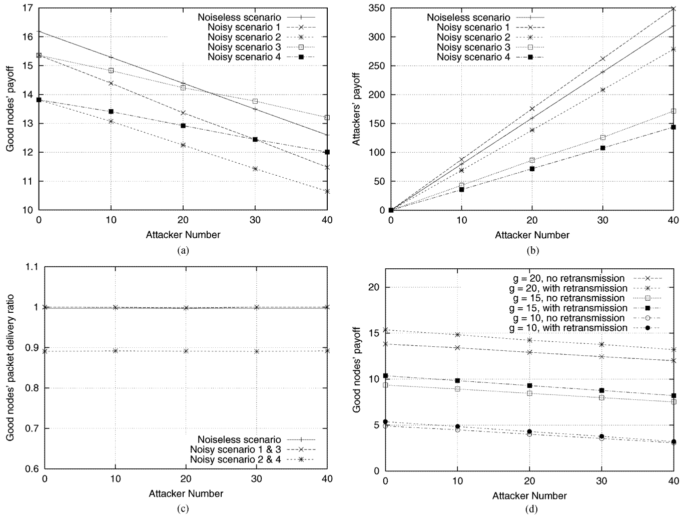


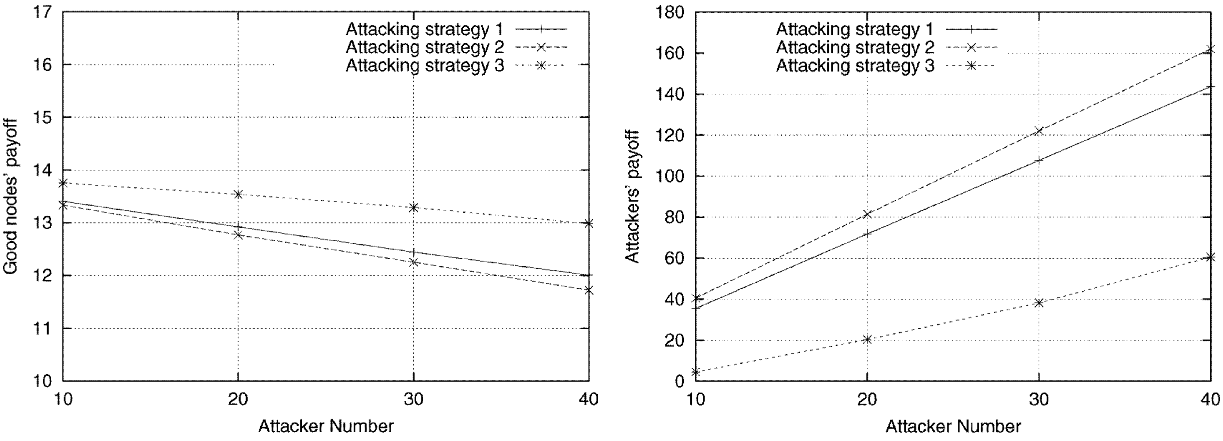
Fig. 5. Payoff comparison when no attackers will drop packets.

Section III-C, and all (insider) attackers follow the optimal attacking strategy described in Section III-D with the only modification being that no attacker will intentionally drop packets. The total simulation time t_f is set to be 10 000 s, and all results are averaged over 20 independent rounds. The following parameters are used: $g = 20$, $c = 1$, $\alpha = 1$, $L_{\max} = 10$, $p_f = 0.05$, $p_m = 0.05$, and $p_s = 0.05$. The acceptable false alarm ratio is set to be 0.1%. For mobile ad-hoc networks, the mobility pattern 4 listed in Table II is used. Since t_f is not very large, p_e is set to be 3%, which is obtained through offline training. For static ad-hoc networks, we focus on the case that the attackers can always find routes with $L_{\max}$ hops to inject packets. For mobile ad-hoc networks, four scenarios are considered, as listed in Table III, and DSR [23] is used as the underlying routing protocol to perform route discovery. The simulation results are illustrated in Fig. 5.

Fig. 5(a) compares the good nodes' payoff under different scenarios. First, we can see that when no attackers are present, the noiseless scenario has the highest payoff, and the noisy scenario 2 and 4 (no retransmission is allowed upon unsuccessful packet delivery) have the lowest payoff. The reason is that the good nodes' payoff is determined not only by their transmission cost, but also by the packet delivery ratio. Under noisy environments, when no retransmission is allowed upon unsuccessful packet delivery, the packet delivery ratio will also be decreased, as illustrated in Fig. 5(a), where, in this case, the packet delivery ratio is only about 89% [illustrated in Fig. 5(c)]. Second, we can see that the allowance of retransmission upon unsuccessful packet delivery can increase the good nodes' payoff in these scenarios (noisy scenario 1 versus noisy scenario 3, and noisy scenario 2 versus noisy scenario 4). However, with the increase of the number of attackers, the performance gap between the two scenarios (with or without retransmission) will also decrease (noisy scenario 1 versus noisy scenario 2, and noisy scenario 3 versus noisy scenario 4). Third, in general, noise will decrease the good nodes' payoff; however, the noisy scenario 3 can achieve higher payoff than the noiseless scenario when the attacker number is no less than 30. The reason is that in the noiseless scenario, attackers can always find $L_{\max}$-hop routes, while in the noisy scenario 3, the average hop number per route selected by the attackers is much less than $L_{\max}$, and the caused damage is less than that in the noiseless scenario.

Fig. 5(b) demonstrates the attackers' payoff under different scenarios. First, as shown in the case of noisy scenario 3 and 4, when the attackers cannot always use $L_{\max}$-hop routes to inject packets, their payoff will be decreased a lot compared to the cases that they can, as shown in the case of noisy scenario 1 and 2. Second, the allowance of retransmission upon unsuccessful packet delivery can also increase the attackers' payoff, since now more packets can be injected by the attackers. Third, since the attackers' packets may also be dropped under the noisy scenarios, without allowing retransmission, the attackers' payoff will also be decreased compared to the noiseless scenario, as shown by the noisy scenario 2. However, when retransmission is allowed, compared to the noiseless scenario, the attackers' payoff can still be increased even under the noisy scenarios, as illustrated by the noisy scenario 1.

Finally, Fig. 5(d) illustrates the good nodes' payoff under different g values, where now only the noisy scenario 3 and 4



(a)

(b)

Fig. 6. Payoff comparison when some attackers will drop packets.

are considered. First, from these results, we can see that with the increase of the number of attackers, the performance gap between these two scenarios will also decrease. The reason is that the attackers can take advantage of retransmission to cause more damage to the good nodes. Second, with the decrease of g, the performance gap between these two scenarios will also decrease. For example, when $g = 10$ and the number of attackers is 40, there is almost no difference. In summary, the gain introduced by the allowance of retransmission becomes less and less with the increase of the number of attackers or with the decrease of g. However, it is worth mentioning that g does not change the underlying strategy design as long as it is reasonably large.

Thus far, we have only considered the situations that no attackers will intentionally drop packets. Next, we study the situation when the attackers will also try to drop the good nodes' packets. In this set of simulations, three attacking strategies will be studied: in "attacking strategy 1," no attackers will intentionally drop the good nodes' packets. In "attacking strategy 2," each attacker will only drop the first B'_{th} packets for any good node that has requested it to forward, then will stop participating route discoveries initiated by that good node, where dropping B'_{th} packets will not be detected as malicious. In these simulations, we set $B'_{th} = 20$. In "attacking strategy 3," each attacker will always keep participating the route discoveries initiated by the good nodes and will drop the good nodes' packets in such a way that it will not be detected as malicious, which can be regarded as selective dropping.

Fig. 6(a) illustrates the good nodes' payoff under different attacks. First, compared to the attacking strategy 1, attacking strategy 3 even increases the good nodes' payoff, though the attackers can drop some good nodes' packets. The reason is that when the attacking strategy 3 is used, the attackers also need to keep forwarding packets for the good nodes, which will increase the number of nodes that the good nodes can use and reduce the value of $\bar{L}_{\min}$. Since the number of packets that the attackers can drop without being detected as malicious is very limited, the extra damage that they can cause is also very limited, and the good nodes' payoff will be increased consequently. Second, compared to the attacking strategy 1, attacking strategy 2 can decrease the good nodes' payoff a little bit due to the extra number of packets that they have dropped. However, since the number of packets that the attackers can drop is always bounded, with the increase of time, the effect of such packet dropping becomes less noticeable.

Fig. 6(b) illustrates the attackers' payoff. First, attacking strategy 2 can increase the attackers' payoff compared to attacking strategy 1. The reason is that the attackers can drop some extra packets without being detected when attacking strategy 2 is used. However, attacking strategy 3 can dramatically decrease the attackers' payoff comparing to attacking strategy 1, the reason is that forwarding packets for the good nodes will also incur a lot of cost, while the number of packets that they can drop without being detected as malicious is very limited. In summary, from the attackers' point of view, when the network lifetime is finite, attacking strategy 2 should be used, while its advantage over attacking strategy 1 is very limited, and will decrease with the increase of network lifetime.

VI. Conclusion

In this paper, we have investigated how to secure cooperative ad-hoc networks against insider attacks under realistic scenarios, where the environment is noisy and the underlying monitoring is imperfect. We model the dynamic interactions between good nodes and attackers in such networks as securing routing and packet forwarding game. The optimal defense strategies have been devised, which are optimal in the sense that no other strategies can further increase the good nodes' payoff under attacks. The maximum possible damage that can be caused by the attackers has also been analyzed. By focusing on the worst case scenario from the good nodes' point of view, that is, the good nodes have no prior knowledge of the other nodes' types while the insider attackers can know who are good nodes, the devised strategies can work well under any scenario. Extensive simulations have also been conducted to justify the underlying assumptions and to evaluate the proposed strategies. The simulation results demonstrate that the proposed defending strategies can effectively secure cooperative ad-hoc networks under noise and imperfect monitoring.

Appendix

Proof: [of Theorem 1]: To show that the proposed strategy profile forms a Nash equilibrium, we only need to show that no player can increase its payoff by unilaterally changing its own strategy.

- P_0's actions when it is good: According to the secure routing and packet forwarding strategy, P_0 will take action R if and only if: 1) the packet to be sent is valid; 2) $n \le L_{\max}$; 3) no nodes on this route have been marked as malicious by P_0; 4) all relay nodes have agreed to be on this route; and 5) this route has the minimum cost among all good routes to P_n known by P_0. First, if P_0 takes action R when the packet to be sent is not valid, the good nodes' payoff cannot be increased, or may even be decreased. Second, if P_0 takes action R when $n > L_{\max}$, P_0 will be marked as malicious by other good nodes and cannot send any packets again, which will decrease the good nodes' payoff. Third, if P_0 takes action R when some nodes have been marked as malicious by P_0 or some nodes do not agree to be the route, then the packet will be dropped by a certain relay node and, consequently, all cost spent to transmit this packet will be wasted, and the good nodes' payoff will be decreased. Fourth, if P_0 takes action R when the selected route does not have the minimum cost among all good routes to P_n known by P_0, then compared to the situation that the good route with the minimum cost is used, some extra cost will be wasted if this route is used instead, which will decrease the good nodes' payoff. Finally, if all of the above conditions are satisfied but P_0 takes action NR, the good nodes' payoff will not increase too, since not sending the packet or sending the packet using a nonminimum cost route can bring no gain or can only bring less gain.
- P_0's decision when it is malicious: According to the optimal attacking strategy, P_0 will take action R if and only if: 1) $F_{P_0}(P_0, t) < f_{P_0}(t)$; 2) $n = L_{\max}$; 3) all relay nodes are good; and 4) all relay nodes have agreed to be on this route. First, if P_0 takes action R when $F_{P_0}(P_0, t) \ge f_{P_0}(t)$ or $n > L_{\max}$, P_0 will be marked as malicious by good nodes and cannot inject any packets again, which will surely decrease the attackers' payoff. Second, if P_0 takes action R when $n < L_{\max}$ or some relay nodes are malicious or some relay nodes do not agree to be on this route, since P_0 can always find a route with $L_{\max}$ hops and with all relay nodes being good, using a suboptimal route surely cannot increase P_0's attack efficiency. Third, if all of those conditions are satisfied but P_0 takes action NR, since the maximum possible damage that can be caused by each packet injecting is $(L_{\max} - 1)c$, the attackers' payoff cannot be further increased either.
- P_i's decision $(0 < i < n)$ when it is good: According to the secure routing and packet forwarding strategy, P_i will take action (A, F) if all of the other nodes on this route have not been marked as malicious by it and $n \le L_{\max}$; otherwise, it will take action (NR, D). When no nodes on this route have been marked as malicious by it and $n \le L_{\max}$, since refusing to be on this route may cause the source to select a route with higher cost and dropping packet will waste other good nodes' cost, both will cause P_i' payoff to be decreased. When some nodes on this route have been marked as malicious by P_i or $n > L_{\max}$, if P_i agrees to be on this route or does not drop the packet, since the packet will finally be dropped by malicious node, all effort that has been spent by good nodes in this subgame will be wasted, which surely cannot increase P_i's payoff either.
- P_i's decision $(0 < i < n)$ when it is malicious: According to the optimal attacking strategy, P_i will always take action (NA, D). We first consider the situation that P_0 is good. If P_i takes action (*A*, *D*), it will be detected as malicious immediately and cannot cause damage to P_0 anymore, which surely cannot increase the attackers' payoff. If P_i takes action (*A*, *F*), this can only contribute to good nodes by helping good nodes forward packets, and cannot increase the attackers' payoff. Meanwhile, taking action (*NA*, *F*) surely cannot cause damage the good nodes, since good nodes will not use P_i to forward packets. Now let us consider the situation that the initiator P_0 is malicious. It is also easy to check that taking action (NA, D) is always a best strategy from the malicious nodes' point of view since P_0 can always find a better route, that is, a route with $L_{\max}$ hops and with all relay nodes being good.

Based on the above analysis, we can see that no player can increase its payoff by unilaterally changing its own strategy. ■

References

[1] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Netw. Mag.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[2] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. 6th Annu. Int. Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 275–283.

[3] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annu. Int. Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 255–265.

[4] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking Computing*, Long Beach, CA, May 2001, pp. 146–155.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," presented at the 8th Annu. Int. Conf. Mobile Computing and Networking, Atlanta, GA, Sep. 2002.

[6] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," presented at the ACM Workshop on Wireless Security, San Diego, CA, Sep. 2003.

[7] I. Aad, J. P. Hubaux, and E. Knightly, "Denial of service resilience in ad hoc networks," in *Proc. 10th Annu. Int. Conf. Mobile Computing and Networking*, Philadelphia, PA, Sep. 2004, pp. 202–215.

[8] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks," in *Proc. 24th Annu. Joint Conf. IEEE Computer and Communications Societies*, Miami, FL, Mar. 2005, vol. 2, pp. 1252–1261.

[9] W. Yu and K. J. R. Liu, "Secure cooperative mobile ad hoc networks against injecting traffic attacks," in *Proc. 2nd Annu. IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks*, Sep. 2005, pp. 55–64.

[10] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," presented at the SCS Communication Networks and Distributed Systems Modeling and Simulation Conf., Jan. 2002.

[11] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," presented at the Int. Conf. Network Protocols, Nov. 2002.

[12] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," presented at the ACM Workshop on Wireless Security, Sep. 2002.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," presented at the IEEE Infocom, 2003.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Netw. J.*, vol. 1, pp. 175–192, 2003.

[15] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Mobihoc*, 2002, pp. 226–236.

[16] P. Michiardi and R. Molva, "Core: A COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks," presented at the IFIP–Commun. Multimedia Security Conf., 2002.

[17] P. Michiardi and R. Molva, "Game theoretic analysis of security in mobile ad hoc networks," Instit. Eurecom, Tech. Rep. RR-02-070, 2002.

[18] S. Buchegger and J.-Y. Le Boudec, "The effect of rumor spreading in reputation systems for mobile ad-hoc networks," presented at the WiOpt, 2003.

[19] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA: MIT Press, 1994.

[20] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," presented at the IEEE INFOCOM, 2003.

[21] W. Yu and K. J. R. Liu, "Attack-resistant cooperation stimulation in autonomous ad hoc networks," *IEEE J. Selected Areas Commun. Special Issue Autonomic Commun. Syst.*, vol. 23, no. 12, pp. 2260–2271, Dec. 2005.

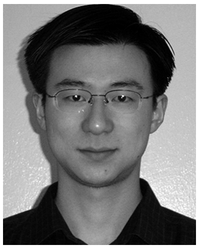
[22] O. Kallenberg, *Foundations of Modern Probability*. New York: Springer-Verlag, 1977.

[23] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," in *Mobile Comput.*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996, ch. 5, pp. 153–181.

[24] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,*, IEEE Std. 802.11-1007, IEEE Comput. Soc. LAN MAN Standards Committee.

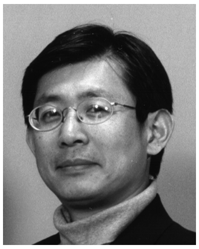
Wei Yu received the B.S. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2000, the M.S. degree in computer science from Washington University, St. Louis, MO, in 2002, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2006.

From 2000 to 2002, he was a Graduate Research Assistant at Washington University. From 2002 to 2006, he was a Graduate Research Assistant with the Communications and Signal Processing Laboratory and the Institute for Systems Research, University of Maryland. He joined Microsoft Corporation, Redmod, WA, in 2006. His research interests include network security, wireless communications and networking, game theory, wireless multimedia, and pattern recognition.



Zhu Ji received the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2007. He received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2000 and 2003, respectively.

Currently, he is with QUALCOMM, San Diego, CA. From 2003 to 2007, he was a graduate research assistant in the Communication and Signal Processing Laboratory, University of Maryland. From 2000 to 2002, he was a Visiting Student (research intern) in the Wireless and Networking Group at Microsoft Research Asia, Beijing. His research interests are in wireless communications and networking.



K. J. Ray Liu (F'03) is Professor and Associate Chair, Graduate Studies and Research, of the Electrical and Computer Engineering Department, University of Maryland, College Park. His research contributions encompass broad aspects of wireless communications and networking, information forensics and security, multimedia communications and signal processing, bioinformatics and biomedical imaging, and signal processing algorithms and architectures. He was the Editor-in-Chief of *IEEE Signal Processing Magazine* and the founding Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing*.

Dr. Liu is Vice President—Publications and on the Board of Governor of IEEE Signal Processing Society. He is the recipient of many honors and awards including best paper awards from the IEEE Signal Processing Society (twice), IEEE Vehicular Technology Society, and EURASIP; IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and National Science Foundation Young Investigator Award. He also received various teaching and research awards from the University of Maryland, including the Distinguished Scholar–Teacher award, Poole and Kent Company Senior Faculty Teaching Award, and the Invention of the Year award.